Emergent Herding Behaviour Through Decentralised Reinforcement Learning

Benjamin Cooper, Khoa Nguyen and Alen Alempijevic
Robotics Institute, University of Technology Sydney, NSW, Australia
{benjamin.j.cooper}@student.uts.edu.au, {khoa.nguyen,alen.alempijevic}@uts.edu.au

Abstract

Herding is a vital aspect of livestock management that requires skilled workers to interpret animal behaviour and respond effectively to dynamic conditions. Although traditionally used for surveillance, drones' mobility and sensing capabilities make them well-suited for autonomous herding. This paper proposes a decentralised reinforcement learning (RL) strategy for a team of aerial agents to herd animals, focusing on emergent behaviour rather than predefined control barrier functions. A biologically grounded Repulsion–Attraction Reward Mechanism and a curriculum learning strategy enable agents to progressively acquire herding skills, starting with spacing and approach, then advancing to encirclement and guidance. Using only local observations and minimal communication, the RL agents exhibit scalable and robust behaviour that mimics natural predator-prey interactions. Simulations demonstrate reliable herding performance across varying behavioural conditions.

1 Introduction

Herding is a critical process in livestock management traditionally performed by skilled humans or trained animals. Effective herding demands an understanding of animal behaviour, adaptability to dynamic conditions, and coordination across terrain [Beaver and Höglund, 2015]. Shepherding requires herders to cooperate and self-organise in order to orchestrate the emergence of desired collective behaviour in the herd whose natural dynamics resist such coordination. This necessity for distributed collaboration and emergent control makes effective shepherding fundamentally dependent on multi-agent systems [Lama and di Bernardo, 2024].

The concept of herding and steering a group of animals using multiple robotic agents was introduced by [Jyh-Ming Lien *et al.*, 2005], demonstrating how coordinated positioning and movement can influence large or difficult-to-control herds. Shepherding presents challenges such as heterogeneous agent dynamics, indirect environmental influence, complex emergent behaviours, and real-time distributed decision-making with limited communication [Napolitano *et al.*, 2025]. Drones offer unique advantages for this task due to their mobility, aerial perspective, and sensing capabilities. Originally used for agricultural surveillance and predator deterrence [Berezina *et al.*, 2024], drones are increasingly recognised as promising agents for autonomous herding. Their ability to cover large and uneven terrain, maintain persistent visual contact, and coordinate from a vantage point makes them particularly well-suited for decentralised multi-agent shepherding.

Reinforcement Learning (RL) offers a powerful framework for autonomous herding, particularly suited to dynamic environments where agent interactions are complex and system dynamics are difficult to model. A compelling example of emergent behaviour in this context is described in [Nalepka *et al.*, 2017], where humans solving a virtual shepherding task began oscillating around targets to contain them—demonstrating that effective strategies can emerge organically through interaction. RL enables similar emergent coordination among artificial agents, especially in multi-agent settings where decentralised policies must adapt under uncertainty. Recent work has shown that RL can effectively guide stochastic agents in shepherding tasks without relying on cohesion assumptions or predefined control structures [Napolitano *et al.*, 2025; Zhi and Lien, 2021]. These studies highlight RL's capacity to learn robust, scalable behaviours in environments where traditional rule-based or model-driven approaches often fall short.

The main contribution of this paper is a biologically grounded reinforcement learning framework for autonomous herding using aerial drones. We introduce a novel Repulsion–Attraction Reward Mechanism and a

curriculum learning strategy that decomposes the herding task into progressive phases, first learning drone spacing and approach behaviour, then advancing to encirclement and guidance. Drones acquire herding skills through interaction, using only local observations and minimal communication, without relying on explicit models, centralised coordination, or predefined control hierarchies. Our approach supports scalable, emergent coordination and offers a possibility to learn from interaction for a model-free solution for autonomous livestock management.

2 Related Work

Approaches to modelling herd behaviour vary significantly. A stochastic, non-cohesive model where agents move independently without alignment or cohesion was proposed by [Napolitano *et al.*, 2025]. Reynolds' Boids model [Reynolds, 1987], is widely used in animal flocking simulations (e.g., [Bhattacharyya and Kennedy, 2022]), introducing rules of separation, alignment, and cohesion to generate emergent group behaviour. However, for controlled herding scenarios, Olfati-Saber's framework [Olfati-Saber, 2006] offers a more suitable alternative by embedding these rules within a control-theoretic and consensus-based structure.

Early robotic herding approaches used deterministic control strategies such as Model Predictive Control (MPC), relying on explicit models of agent dynamics and herd behaviour [Pierson and Schwager, 2018]. To improve scalability and decentralisation, Distributed MPC (DMPC) frameworks were introduced, where local control actions are computed using only neighbourhood information, often through iterative optimisation schemes such as decentralised sequential quadratic programming [Stomberg *et al.*, 2023]. While DMPC improves scalability and reduces reliance on centralised coordination of MPC approaches, it still depends on synchronised communication and predefined formations, limiting flexibility.

To address limitations of predator-prey heuristics and improve convergence, several works reformulated herding as a multi-robot formation control problem. These approaches use virtual linkages, potential fields, or Control Barrier Functions (CBFs) to enforce safe spacing and coordinated encirclement of the herd [Chipade and Panagou, 2021; Chen *et al.*, 2017]. While effective in maintaining formation and avoiding collisions, such methods often assume access to global state information and require tight inter-agent coordination. Moreover, they typically encode behaviour through handcrafted rules or constraints, limiting adaptability to diverse herd responses and terrain conditions.

Attraction-repulsion dynamics, have inspired the framework proposed in [Nguyen *et al.*, 2024], where inter-agent forces are used to maintain optimal spacing and prevent collisions during herding. By modelling agents with both attractive and repulsive forces, the system achieves emergent coordination without centralised control. The equilibrium point of the interaction function defines the desired inter-agent distance, allowing agents to self-organise into effective formations. This approach struggles with smaller groups due to lattice formation issues and herd fragmentation. In contrast, learning-based method could adapt to varying herd sizes and drone configurations, offering improved scalability and robustness in dynamic environments.

Recent work has explored reinforcement learning (RL) as a model-free alternative for multi-agent herding, enabling agents to learn control policies directly from interaction data and relaxing assumptions about herd cohesion and agent determinism [Napolitano *et al.*, 2025]. Hierarchical RL frameworks have been proposed to separate high-level decision-making from low-level control, and some rely on perceptual-motor primitives or centralised training pipelines. However, these approaches often impose structural constraints or lack scalability in decentralised settings. More broadly, RL offers a flexible framework for autonomous agents operating in dynamic environments, allowing them to adapt through experience rather than relying on predefined models or heuristics. Multi-Agent Reinforcement Learning (MARL) extends these benefits to multi-agent systems, supporting decentralised policy learning under uncertainty and partial observability. Prior work has demonstrated MARL's effectiveness in tasks such as multi-agent path planning and coordination in dynamic landscapes [Hu *et al.*, 2025; Zhang and Liu, 2023; Liu *et al.*, 2020], highlighting its potential to outperform traditional rule-based or model-driven approaches in scenarios where system dynamics are difficult to model or predict.

3 Methodology

3.1 Flocking behaviour

In this work, cattle behaviour is modelled using Olfati-Saber's multi-agent flocking framework [Olfati-Saber, 2006]. Within this framework, each agent (cow) aligns its movement with neighbouring agents while maintaining separation and cohesion, resulting in emergent flocking behaviour. By employing the α-lattice structure, the model ensures that the herd maintains a coherent formation while still allowing individual agents to respond dynamically to external stimuli, such as the influence of herding drones. The flocking control is defined in (1). The first term is a gradient-based term that drives agents to maintain appropriate spacing from neighbours, using the function ϕ_α and normalised $\mathbf{n}_{ij}$. The second term is a consensus term that aligns the velocities of neighbouring

agents through the adjacency-like coefficients $\mathbf{a}_{ij}$. Together these terms generate realistic flocking behaviour that balances cohesion, separation and velocity alignment.

$$u_i^{\alpha} = \underbrace{\sum_{j \in N_i} \phi_{\alpha}(\|q_j - q_i\|_{\sigma})\mathbf{n}_{ij}}_{\text{gradient-based term}} + \underbrace{\sum_{j \in N_i} a_{ij}(q)(p_j - p_i)}_{\text{consensus term}} \quad (1)$$

This formulation allows the herd to naturally form dynamic, yet structured, patterns in free space while remaining responsive to external control inputs. Such a model provides a foundation for simulating realistic herding scenarios, where UAVs can influence the herd without rigidly prescribing trajectories. Such an approach provides a realistic and flexible environment for training and evaluating multi-agent herding strategies.

3.2 Model

This study employed the Proximal Policy Optimisation (PPO) algorithm from the Stable-Baselines3 library [Raffin *et al.*, 2021] as the reinforcement learning framework. PPO, an advancement over Trust Region Policy Optimisation (TRPO) [Schulman *et al.*, 2017], uses a clipping mechanism to ensure stable learning. For multi-agent coordination, we adopted a Multi-Agent PPO (MAPPO) structure, leveraging a centralised training, decentralised execution (CTDE) paradigm [Amato, 2024]. This allowed agents to benefit from shared training information while maintaining decentralised observation inputs during execution. The simulation environment was built on a modified version of the gym-pybullet-drones framework [Panerati *et al.*, 2021], tailored to support dynamic cattle herding scenarios with multiple UAV agents.

3.3 Rewards

Rewards were critical to shaping the RL model and how it behaves. The model received rewards that encouraged the model to take appropriate actions that would lead it to herding the cattle and was penalised when taking an action that contradict this goal. The model was rewarded for moving the centroid of the drones closer to the centroid of the cattle. The centroid of each group was computed based on the average x,y positions of each agent for their respective swarm. This reward encouraged the drones to work together to move the centroid of the drones over to the same spot of the centroid of the cattle.

The model then received two rewards based on spacing to promote the correct interaction behaviour between drones and other drones and drones to cattle. The reward for spacing operated on the principle of maintaining a desired distance: Give a reward for getting closer to the desired distance but be penalised for getting too close. This reward structure is discussed further in the next section. All of these rewards where normalised to produce a reward between [-1, 1]. After each reward was normalised they each received a weighting to help the model distinguish which reward were more important and critical to successfully herding the cattle.

3.4 Repulsion–Attraction Reward Mechanism

One of the key contributions of this paper is the development of a unified reward function tailored for reinforcement learning, building upon the attraction–repulsion formulation introduced in [Nguyen *et al.*, 2024]. In that framework, agents experience strong repulsion when too close and moderate attraction when far apart, with an equilibrium point representing the desired inter-agent distance. While effective in rule-based systems, this formulation poses challenges for reinforcement learning: directly using repulsion as a negative reward and attraction as a positive reward can lead agents to maximise attraction by maintaining excessively large distances. To address this, we propose a single, piecewise reward function that balances short-range precision with long-range guidance:

$$f(r) = \begin{cases} A\exp\left(-\frac{(r-d)^2}{2c^2}\right) - B\exp\left(-\frac{r^2}{2k^2}\right), & r \le r_0, \\ C\exp(-\lambda r), & r > r_0, \end{cases} \quad (2)$$

This formulation combines two Gaussian terms in the short-range regime ($r \le r_0$) to create a reward landscape with a positive peak at the desired distance ddd and a sharp penalty for proximity violations. For larger distances ($r > r_0$), an exponential decay term ensures that agents still receive a diminishing incentive to approach the cattle, preventing disengagement at long range. The resulting reward encourages agents to maintain an optimal distance from the cattle, penalises unsafe proximity, and provides consistent directional guidance. Parameters were tuned to normalise the reward between approximately -1 and +1, making it suitable for reinforcement learning.

The C from (2) calculated using (3) to ensure a smooth continuation between the two functions. The reward function of (2) is shown in Figure 1.

$$C = \frac{A\exp\left(-\frac{(r_0-d)^2}{2c^2}\right) - B\exp\left(-\frac{r_0^2}{2k^2}\right)}{\exp(-\lambda r_0)} \quad (3)$$

The second reward function modifies the initial design to better guide drone-to-drone interactions in reinforcement learning. A key change was replacing the

long-distance exponential reward with a linear penalty, discouraging drones from spreading too far apart. Unlike the first reward, which allowed drones to be distant from cattle without penalty, this formulation ensures drones remain close to each other at the start, regardless of cattle position. Additionally, the negative Gaussian component was replaced with a linear relationship, simplifying the tuning process and improving training stability.

$$R(r) = G(r) + C(r) + L(r) \tag{4}$$

The overall expression for the drone-to-drone reward function R(r) is given (4). This function is composed of three components: G(r) provides a positive reward for approaching the desired inter-drone distance; C(r) penalises drones for getting too close; and L(r) penalises for being too far apart. These are described in (5), (6) and (7) respectively.

$$G(r) = \exp\left[-\tfrac{1}{2}\left(\frac{r - d_\star}{\sigma}\right)^2\right] \tag{5}$$

$$C(r) = \begin{cases} -P_{\text{coll}}\left(1 - \dfrac{r}{r_{\text{coll}}}\right), & r < r_{\text{coll}}, \\ 0, & r \geq r_{\text{coll}} \end{cases} \tag{6}$$

$$L(r) = \begin{cases} 0, & r \leq r_L, \\ -\dfrac{r - r_L}{r_{\max} - r_L}, & r > r_L \end{cases} \tag{7}$$

Where r_{coll} is the threshold distance for the collision penalty, and r_{L} is the threshold distance at which the long-range attraction term begins to apply. The reward function of (4) is shown in Figure 2.

3.5 Curriculum Learning

To facilitate the progressive development of agent behaviours, curriculum learning was employed to structure the training of the PPO model. This approach enabled agents to first master foundational skills before advancing to more complex tasks. The curriculum was divided into seven levels, targeting three core behavioural domains: formation, approach, and herding. The formation phase consisted of three levels, each imposing increasingly strict constraints on drone-to-drone spacing. The approach phase included two levels focused on teaching agents how to effectively approach and begin surrounding cattle. Finally, the herding phase comprised two levels aimed at refining cattle surrounding techniques and maintaining optimal cattle spacing. Across curriculum levels, reward functions were reweighted and termination conditions adjusted to align with the desired behaviours at each stage.

3.6 Agent Observation Space

Each drone received observation data comprising of its own state and the relative positions of nearby agents. The drone's internal state was represented by a 10-dimensional vector including altitude, orientation (roll, pitch, yaw), linear velocity, and angular velocity. To maintain decentralised execution, the positions of neighbouring drones and cattle were encoded as 2D displacement vectors (x, y) relative to the observing drone, omitting the vertical (z) component to align with the 2D nature of the flocking behaviour model. The observation data the agent received was normalised.

To investigate the scalability and adaptability of the reinforcement learning model, the number of drones per episode varied between 4 and 12. Since varying agents would typically result in changing observation sizes, unsuitable for standard RL training, the observation space was constrained by limiting each drone to data from its four nearest neighbours, forming a localised neighbour-to-neighbour topology. This design enabled the model to generalise across dynamic team sizes. Formally, the total observation vector for each drone was computed as:

$$\begin{aligned} \text{obs\_length} = {} & \text{own\_state\_vector\_length} \\ & + (\text{max\_neighbour\_drones} \times 2) \\ & + (\text{max\_nearby\_cattle} \times 2) \\ & + (\text{action\_buffer\_size} \times \text{action\_space}) \end{aligned} \tag{8}$$

A key assumption in this work was that the global positions of all cattle are known, drones only received a normalised 2D vector pointing towards each nearby cow. This ensured that decision were made based on local, relative observation while still allowing drones to coordinate effectively in a herding task.

4 Experiments

To evaluate the performance of the proposed herding RL model, experiments were conducted in a simulated environment using pybullet, designed to replicate key aspects of multi-agent cattle herding. In each episode the herd was randomly spawned within the environment, introducing variability in initial conditions and ensuring that agents could not over-fit to a fixed starting configuration. Additionally the number of drones available for herding varied between episodes from 4-12 drones. This allowed us to assess the overall robustness and scalability of the learned policies under different levels of agent resources.

4.1 Reward Parameters

Each reward in the learning framework was weighted according to its influence on agent behaviour. Centroid distance received the highest weighting (1.0), as minimising this distance was critical for ensuring drones effectively

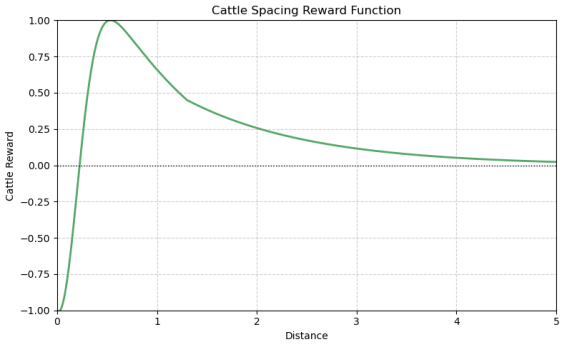

Figure 1: Repulsion–attraction reward mechanism based for drone to cattle spacing

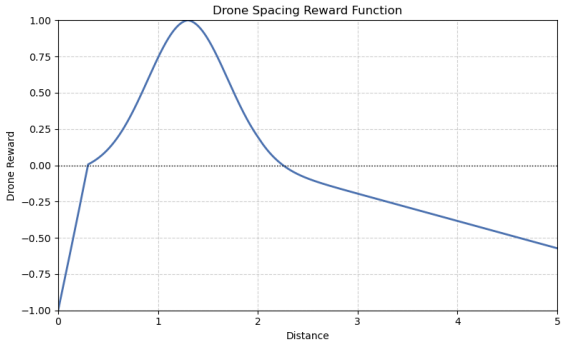

Figure 2: Repulsion–attraction reward mechanism based on drone to drone spacing

approach and engage with the cattle herd. Drone-to-drone spacing was weighted at 0.8 to promote coordinated movement and prevent clustering, while drone-to-cattle spacing was assigned a lower weight (0.6). Prioritising drone-to-drone spacing over drone-to-cattle spacing helped avoid scenarios where multiple drones converged on the same animal, which could lead to inefficient herding and agent conflict. This reward structure encouraged distributed coverage of the herd and improved overall coordination among agents

The repulsion–attraction reward mechanism (2) for drone-to-cattle spacing was calibrated to promote the RL model to learn the graph correctly. The following parameter where used: positive amplification coefficient $A = 1.5$, negative amplification coefficient $B = 2.1$, width of the positive Gaussian $c = 3.3$, width of the negative Gaussian $k = 0.3$, positive peak offset d $= -1$, piecewise threshold $R_0 = 1.3$, and exponential decay $\lambda = 0.8$. These parameters tuned the function to produce rewards around the range -1 to $+1$, and resulted in agents learning to maintain a desired inter-agent distance of approximately 0.75 units. Figure 1 shows the graph with these parameters.

The reward function for the drone-to-drone spacing, as described in equation 4, was given the following parameters: desired distance $d = 0.8$, $\sigma = 0.25$, $r_{\text{coll}} = 0.3$, and $r_{\text{L}} = 1.5$. Figure 2 shows the resultant graph.

When a training episode was successfully terminated, the agent received a bonus reward composed of two part. The first component depended on the distance between the drone centroid and the cattle centroid: a positive reward was given if the drones reached the cattle centroid; otherwise, a penalty equal to 1.5 times the distance was applied.. The second component was based on herding success: a large bonus was awarded for herding all cattle, a penalty for herding none, and a proportional reward for partial success based on the number of cattle gathered.

4.2 Evaluation

The RL model was evaluated in a simulated environment under varying conditions to assess its robustness. Evaluation data was collected across 100 episodes, the cattle were randomly spawned in different positions within the environment, ensuring that agents were tested against diverse initial configurations. To further examine scalability, the number of drones available for herding varied between 4 and 12, altering the drone-to-cattle ratio. To demonstrate the RL models performance the simulator used in [Nguyen *et al.*, 2024] was also evaluated under the same conditions with 100 simulations recorded for comparison.

Performance of CBF and RL was predominantly measured using an effectiveness score, representing the percentage of cattle successfully encircled by the drone. To calculate this, a winding number algorithm was used. The drones' positions formed a polygon, and each cattle's location was checked to determine whether it lay within this polygon. Additionally, a time analysis was conducted to compare how many steps each method required to complete the encirclement.

This evaluation framework provided a comprehensive view of the models capabilities, measuring not only the effectiveness of the herding strategy but also the efficiency and scalability of the learned polices across different levels of agent resources.

5 Results

A statistical summary of the performance of the CBF and RL operating with 4, 6, 8, 10 and 12 drones is shown in Figure 3. These results indicate that RL exhibited more consistent performance across varying drone counts, albeit at the expense of overall effectiveness. In contrast, CBF demonstrated more variable performance, with noticeable fluctuations depending on the number of drones.

The most significant difference between the two methods lies in the reasons for failing to encircle cattle or

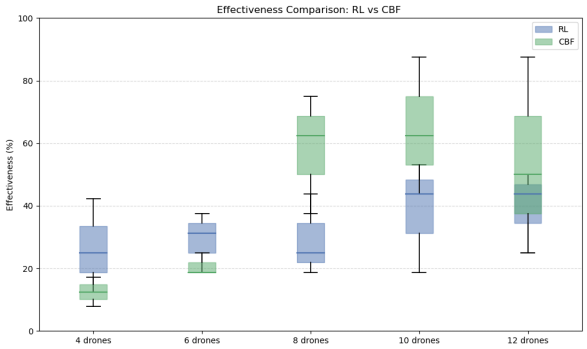

Figure 3: Comparison of CBF and RL effectiveness based on summary statistics (mean, median, and quartiles)

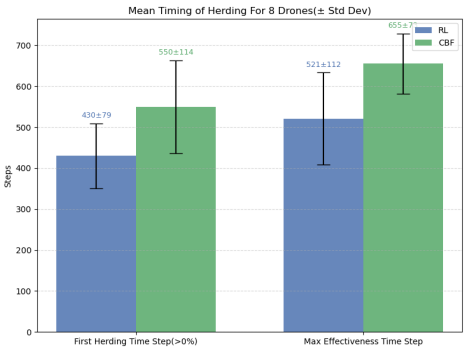

Figure 4: Comparison of mean time steps for CBF and RL methods in the 8-drone herding scenario. The average time step at which each method first achieved a non-zero score and when the peak score was reached

achieve a perfect score of 100. For CBF, the primary limitation was cattle fragmentation. As drones approach the herd, it was common for two or more cattle to escape before the encirclement began. If too many cattle broke away and formed a secondary herd, the CBF system often became trapped between the two groups, unable to proceed effectively. In contrast, RL's main failure mode was drone-to-drone collisions, which frequently led to early episode termination. For instance, in the 8-drone scenario, RL episodes resulted in zero scores due to collisions in 29 out of 100 samples. A critical distinction between the two methods is safety. CBF, with its hard safety constraints, was never observed to enter an unsafe state. RL, however, did violate safety constraints. During curriculum training, the initial phase focused on teaching drones to maintain spacing. While the RL model initially succeeded, its spacing performance deteriorated as additional reward components were introduced with higher weightings.

Figure 4 illustrates the average time steps at which CBF and RL methods first achieved a non-zero score and reached peak performance in the 8-drone herding scenario. RL transitioned from initial coordination to peak effectiveness in about 90 steps, while CBF required approximately 105 steps, indicating RL's faster convergence despite differences in strategy.

We evaluated the spatial behaviour of the drones during herding episodes. The top row of Figure 5 shows the trajectories of drones and cattle over time, from left to right. Initially, the drones start dispersed and positioned ahead of the herd. As the episode unfolds, they move in coordinated fashion towards the cattle, gradually influencing the herds motion. In the later stages, the drones settle form a rough enclosing front that guides the cattle. This emergent behaviour suggests that the RL policy enables a decentralised yet coordinated drone control.

In addition to showing the movement of the drones, the second row of graphs from Figure 5 illustrates model performance during an episode. The centroid distance graph tracks the distance between the centroids of the herding drones and that of the cattle. The decreasing trend over time demonstrates the RL model successfully learned to respond to the centroid-based reward, guiding the drones to coordinate and bring the drone and cattle centroids closer together.

The distribution range graph depicts the maximum radial distance of drones and cattle from their respective centroids. The cattle distribution initially expands as they react to the approaching drones before contracts and matches the drone distribution. Conversely, the drones distribution narrows as they move closer to the herd indicating a collective tightening of their formation before increasing towards the end as drones spread to maintaining their spacing.

The drone evenness plot shows the variance in the mean distance from each drone to its two nearest cattle. High variance early in the episode reflects uneven deployment, which rapidly decreases as the drones spread more uniformly. After 700 steps, the variance remains low with minor fluctuations, indicating stable convergence onto the herd.

The centroid-distance convergence time box plot summaries the distribution of time steps required across simulation runs to reach a stable centroid distance. Most runs converge within 200-600 steps, though a few outliers take up to 1,500 steps. This variability reflects differences in the initial spatial configurations of the drones and cattle.

It can be observed that each drone is able to maintain appropriate spacing with other drones while collectively approaching the herd. These results demonstrate the models capacity to learn in a changing and dynamic environment, highlighting the potential of RL to prove a more adaptive and flexible approach. Such an approach can better respond to environmental variability and can be scaled to coordinate larger numbers of drones and cattle.

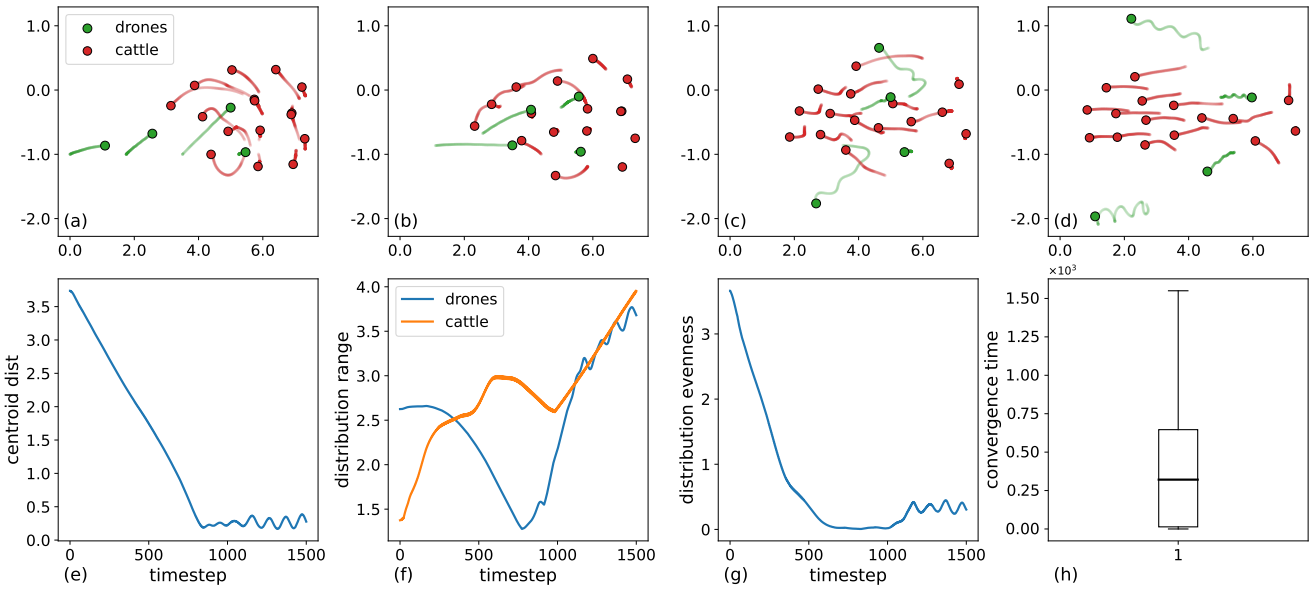

Figure 5: Spatio-temporal evolution and performance metrics of the surrounding task. The top row (a–d) shows the spatial distribution of drones (green) and cattle (red) across four sequential segments of an episode. The bottom row (e–h) presents quantitative measures: (e) distance between drone and cattle centroids over time, (f) distribution range of each group relative to its centroid, (g) evenness of drone positions with respect to nearest cattle, and (h) convergence time across episodes.

6 Discussion and Future Work

This study introduced a novel emergent herding behaviour using decentralised reinforcement learning (RL), demonstrating its adaptability to dynamic environments and stochastic cattle movement. Two reward functions were developed to guide agent interactions, and the observation space design enabled scalability across 4–12 drones. These results highlight the potential of RL for modelling complex, real-world herding scenarios, offering superior scalability over rule-based methods. While CBF doesn't share the same scalability of RL it offered the safest solution and generally more consistent performance.

A key limitation of RL observed in this work was its poor safety performance, underscoring the need for integrated safety mechanisms during training As shown in the works of [Cheng *et al.*, 2019] combining safety though the means of shielding or using CBF and developing a safe reinforcement learning model (SRL) can improve the results of RL while ensuring end-to-end safety.

Future work should focus on enhancing robustness and realism. The current model does not account for adversarial cattle behaviour, which can deviate from typical patterns. Incorporating such dynamics—e.g., cattle that panic, resist movement, or behave unpredictably would provide a deeper evaluation of RL and CBF adaptability. Additionally, scaling the model to larger herds is essential for real-world deployment and offers further opportunities to assess coordination across varying agent-to-cattle ratios.

Acknowledgements

This work is supported in part by Australian Government Research Training Program (RTP) Scholarship and the University of Technology Sydney.

References

[Amato, 2024] Christopher Amato. An introduction to centralized training for decentralized execution in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2409.03052*, 2024.

[Beaver and Höglund, 2015] Bonnie V Beaver and Don Höglund. *Efficient livestock handling: the practical application of animal welfare and behavioral science*. Academic Press, 2015.

[Berezina *et al.*, 2024] EA Berezina, AN Giljov, and KA Karenina. The use of drones for studying the behavior of mammals. *Biology Bulletin*, 51(9):2960–2976, 2024.

[Bhattacharyya and Kennedy, 2022] Dashiell Bhattacharyya and William G. Kennedy. Flocking with only two parameters. In *Conference of The Computational Social Science Society of the Americas*. Springer, 2022.

[Chen *et al.*, 2017] Yu Fan Chen, Miao Liu, Michael Everett, and Jonathan P How. Decentralized non-communicating multiagent collision avoidance with

deep reinforcement learning. In *IEEE International Conference on Robotics and Automation*, pages 285–292. IEEE, 2017.

[Cheng *et al.*, 2019] Richard Cheng, Gábor Orosz, Richard M. Murray, and Joel W. Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. *arXiv preprint arXiv:1903.08792*, 2019.

[Chipade and Panagou, 2021] Vishnu S. Chipade and Dimitra Panagou. Multiagent herding using control barrier functions and input-to-state safe control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 11163–11169. IEEE, 2021.

[Hu *et al.*, 2025] Siyi Hu, Mohamad A. Hady, Jianglin Qiao, Jimmy Cao, Mahardhika Pratama, and Ryszard Kowalczyk. Adaptability in multi-agent reinforcement learning: A framework and unified review. *arXiv*, 2025.

[Jyh-Ming Lien *et al.*, 2005] Jyh-Ming Lien, S. Rodriguez, J. Malric, and N.M. Amato. Shepherding Behaviors with Multiple Shepherds. In *IEEE International Conference on Robotics and Automation*, pages 3402–3407. IEEE, 2005.

[Lama and di Bernardo, 2024] Andrea Lama and Mario di Bernardo. Shepherding and herdability in complex multiagent systems. *Physical Review Research*, 6(3):L032012, 2024.

[Liu *et al.*, 2020] Zuxin Liu, Baiming Chen, Hongyi Zhou, Guru Koushik, Martial Hebert, and Ding Zhao. Mapper: Multi-agent path planning with evolutionary reinforcement learning in mixed dynamic environments. *arXiv*, 2020.

[Nalepka *et al.*, 2017] Patrick Nalepka, Rachel W Kallen, Anthony Chemero, Elliot Saltzman, and Michael J Richardson. Herd those sheep: Emergent multiagent coordination and behavioral-mode switching. *Psychological science*, 28(5):630–650, 2017.

[Napolitano *et al.*, 2025] Italo Napolitano, Stefano Covone, Andrea Lama, Francesco De Lellis, and Mario di Bernardo. Hierarchical learning-based control for multi-agent shepherding of stochastic autonomous agents. *arXiv preprint arXiv:2508.02632*, 2025.

[Nguyen *et al.*, 2024] Dac Dang Khoa Nguyen, Gavin Paul, and Alen Alempijevic. Decentralized multi-phase formation control for cattle herding. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 17948–17953, 2024.

[Olfati-Saber, 2006] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[Panerati *et al.*, 2021] Jacopo Panerati, Hehui Zheng, SiQi Zhou, James Xu, Amanda Prorok, and Angela P. Schoellig. Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7512–7519, 2021.

[Pierson and Schwager, 2018] Alyssa Pierson and Mac Schwager. Controlling noncooperative herds with robotic herders. *IEEE Transactions on Robotics*, 34(2):517–525, 2018.

[Raffin *et al.*, 2021] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[Reynolds, 1987] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *Conference on Computer Graphics and Interactive Techniques*, pages 25–34. ACM, 1987.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov Openai. Proximal policy optimization algorithms. 7 2017.

[Stomberg *et al.*, 2023] Gösta Stomberg, Henrik Ebel, Timm Faulwasser, and Peter Eberhard. Cooperative distributed MPC via decentralized real-time optimization: Implementation results for robot formations. *Control Engineering Practice*, 138:105579, 2023.

[Zhang and Liu, 2023] H. Zhang and Z. Liu. A survey on multi-agent reinforcement learning and its application. *Journal of Artificial Intelligence*, 1(1):1–18, 2023.

[Zhi and Lien, 2021] Jixuan Zhi and Jyh-Ming Lien. Learning to herd agents amongst obstacles: Training robust shepherding behaviors using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 6(2):4163–4168, 2021.