Cooperation or Defection: Multi-Agent Learning in a 3D Continuous Social Sequential Dilemma Game

Martin Le Bras (389340), Andrea Miele (302925), Jean Siffert (366682), Tom Stanic (403843)
CS-503 Project report

***Abstract*—This project extends Multi-agent Reinforcement Learning (MARL) approaches to Sequential Social Dilemmas (SSDs) by transitioning from discrete to continuous action spaces in 3D in the context of the Wolfpack game. The environment's realism is enhanced by realistic vision, multimodal sensory input, LSTM-based memory systems and learnable prey policy. We investigate how these more realistic conditions affect emergent behaviors (defection or cooperation) compared to traditional grid-based simulations. [1]**

Link to Github repository

I. Introduction

Multi-Agent Reinforcement Learning (MARL) has shown promising results in the study of sequential social dilemmas (SSD) where agents must learn to cooperate or defect simultaneously over time. These dilemmas are commonly modeled in simplified grid-based environments with discrete action spaces [2], [3]. However those approaches may overlook some key aspects of agent behavior in more complex and realistic scenarios. The paper [1] introduces the *Wolfpack* game, a cooperative hunting scenario in which two predators (wolves) have to hunt and capture a prey (goat). Based on empirical observations that pack hunting improves carcass defense against scavengers and increases hunting efficiency, the environment is designed to incentivize cooperative behavior among agents, thereby giving rise to a sequential social dilemma. However, the original implementation relies on a grid-based environment with a discrete action space. In our work, we extend this framework to a 3D continuous action space using Unity, incorporating realistic perception through vision and multimodal sensory inputs, as well as memory mechanisms. This setup highlights the sequential and temporally extended nature of decision-making, where agents must not only decide whether to cooperate or defect but also learn how to implement these strategies over time.

The core objective of our project is to model the Wolfpack game in a more realistic setting and analyze the dynamics of cooperation and defection in sequential settings using MARL. Specifically, we explore how increasing the environment realism influences emergent behaviors. First, we assess the impact of transitioning from a discrete to a continuous action space on agents ability to develop coordinated behaviors. Second, we examine how learning dynamics are influenced by enhanced perceptual capabilities, including various vision parameters, integrated memory mechanisms, and additional sensory modalities such as smell.

Addressing these questions is important to assess whether current analyses of SSDs, which have been conducted primarily in simplified grid-like environments, remain relevant when adding complexity and realism to the environment. SSDs provide a more realistic treatment of Matrix Game Social Dilemmas (MGSD) by taking into account the sequential structure [1] and adding even more realism using elements presented above decreases the gap between simplified simulations and real-word applications. The added realism may reveal new dynamics or challenge existing assumptions about cooperative behavior. Understanding how these findings translate to more complex settings is crucial for applying MARL solutions to real-world scenarios where decisions are rarely discrete and environmental information is noisy and multimodal.

II. Related works

Base paper: Our framework builds upon the sequential social dilemma (SSD) paradigm introduced by Leibo *et al.* [1]. In their "Wolfpack" game, two predators and one prey interact in a 2D continuous environment, receiving a solo reward r_s for individual captures and a larger team reward r_t when both predators capture the prey within a capture radius r_c. Varying the ratio r_t/r_s and r_c yields emergent cooperative or defective behaviors. They quantify defection via the lone-wolf capture metric

Multi-Agent Reinforcement Learning: Our work lies within the broader field of multi-agent reinforcement learning (MARL), where agents learn to coordinate or compete in shared environments. Early methods such as independent Q-learning treat each agent as a self-contained learner but often suffer from non-stationarity [4]. Centralized training with decentralized execution (CTDE) approaches—e.g., QMIX [5] and Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [6]—mitigate this by leveraging global critics during training while preserving local policies at test time.

Vision-Based and 3D Environments: Most early SSD and MARL benchmarks operate in low-dimensional or 2D spaces. In contrast, vision-based RL in 3D simulators such as DeepMind Lab [7], VizDoom [8], and Unity ML-Agents

[9] has demonstrated the feasibility of end-to-end pixel-to-action learning. These environments leverage convolutional architectures [10] and recurrent modules [11] to handle partial observability and temporal dependencies. However, few studies have combined SSD-style social dynamics with full 3D physics and vision-based observations, motivating our extension.

Physics-Based Multi-Agent Tasks: Physics simulators such as MuJoCo [12] have enabled continuous-control benchmarks like Multi-Agent MuJoCo (e.g., cooperative locomotion). Methods including population-based training [13] has been applied to these domains, but often focus on collaborative rather than mixed cooperative–competitive settings. Our work introduces a full 3D Wolfpack environment with MuJoCo-style dynamics and vision inputs, bridging the gap between social-dilemma research and high-fidelity physical simulation.

III. Method

To conduct our research, we began by implementing the Wolfpack game within the Unity engine. This involved creating a 3D environment featuring natural elements like rocks and trees shown in Figure 1, and developing the game logic through C# scripts integrated directly into the simulation.

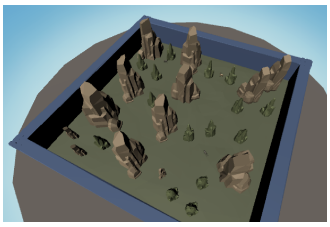


Figure 1: Unity environment for the Wolfpack game

A. Predator vision

Each predator is equipped with ray-casting-based sensors to perceive its surroundings. Ray-casting works by emitting multiple rays in different directions around the agent, with each ray returning information about whether it made contact, the type of object it hit (prey, obstacle, another predator) and the relative distance to that object. We used a vision angle of 250 degrees to approximate the natural field of view observed among wolves [14]. The ray length is set to 30 units, scaled appropriately to match the dimensions of the environment (40% of map size).

B. Prey heuristic

The prey is controlled by a heuristic policy rather than being trained. Its movement consists of randomized translation and rotation (see video [15]), combined with an evasion mechanism (see video [16]). When a predator enters a predefined radius of 10 units, the prey adjusts its trajectory to move away from the predator in an attempt to avoid capture. To allow predators sufficient time to learn basic pursuit and capture strategies, the prey's evasion behavior is enabled only after 7.5M training steps (75% of training). This curriculum-based activation ensures that the predators first learn fundamental hunting skills before encountering more challenging prey behavior.

C. Reward shaping

To implement the Wolfpack game within a reinforcement learning setup, predators receive a reward r_{solo} when capturing the prey individually, and a higher group reward $r_{\text{team}} = k \times r_{\text{solo}}$, $k \geq 1$, when the capture is performed in pack. The group reward is thus scaled proportionally to the solo reward to incentivize cooperative behavior. To promote realistic and efficient movement, predators incur penalties upon collision with obstacles (e.g. rocks, walls) of $r_{collision} = -0.5$. Additionally, a small per-step penalty $r_{time} = -0.02$ is applied throughout each episode. This temporal cost discourages unnecessarily prolonged pursuits and promotes more efficient hunting. Finally, each training episode has a maximum length of 10,000 steps and the predators receive an additional timeout penalty $r_{timeout} = -10$ when they fail to catch the prey.

D. Predator policies

Predator agents are trained within a MARL framework, where each predator maintains an independent policy optimized using the Proximal Policy Optimization (PPO) algorithm [17]. To enhance sample efficiency and stabilize learning, training is conducted concurrently across 4 identically configured map instances, thereby increasing the diversity of observations encountered during policy updates. Agents are trained for 10M steps in episodes consisting of at most 10,000 steps. An episode terminates either upon a successful capture of the prey by a predator or when the step limit is reached without a capture. At the start of each episode, both predators and the prey are randomly initialized at obstacle-free positions on the map. This randomization prevents agents from merely memorizing fixed trajectories to locate the prey.

IV. Experiments

A. Metrics

To quantify emergent cooperation and defection, we employ six metrics:

- **Lone wolf capture rate:** Fraction of captures executed by a single predator.
- **Prey survival time:** Number of timesteps until prey capture or episode timeout.
- **Average inter-predator distance:** Mean predator separation at capture.
- **Proximity rate:** Percentage of timesteps both predators remain within a threshold distance.

- **Capture rate:** Proportion of episodes resulting in prey capture before the maximum timestep.
- **Manual analysis:** Qualitative inspection of trajectories to reveal subtle coordination strategies.

In the analysis of SSDs within the Wolfpack game, the key metric is the *lone-wolf capture rate* L which is the proportion of captures involving only a single wolf. A lower lone-wolf capture rate indicates a higher level of cooperation among predators. This metric is defined as :

$$L = 2 - \frac{1}{C}\sum_{c=1}^{C} w_c \tag{1}$$

where C is the total number of capture and w_c is the number of predators involved in the capture c.

B. Experimental Setup

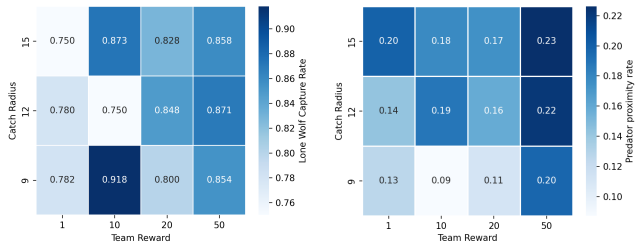
For each experiment, we identified a set of key parameters to explore, along with their respective value ranges. We trained our agents using these parameters and saved their final weights. These weights were then loaded into the agents within an inference environment, which is structurally similar to the training ones. We ran 250 episodes and computed the relevant performance metrics at the end of each episode.

C. Vision reward

Initial experiments revealed that our default reward configuration inherently discouraged cooperative behavior. Indeed the time penalty incentivized predators to capture the prey as quickly as possible, favoring individual over collaborative strategies, even when the group reward was substantially higher. To address this, we introduced an additional positive reward r_{vision} whenever a predator sees the prey (hit by a ray). This modification encourages predators to actively search for and track the prey without necessarily capturing it, creating opportunities for cooperation and pack hunting. We experimented with several values for $r_{\text{vision}} \in \{0, 0.02, 0.04, 0.06\}$, which, accounting for the existing time penalty, resulted in net step rewards of $\{-0.02, 0, 0.02, 0.04\}$ whenever the prey was in sight. Among these, $r_{\text{vision}} = 0.06$ consistently led to more cooperative behaviors and improved task performance. Based on these observations, we adopted this setting as the new baseline, as it effectively promotes proactive pursuit (keeping prey in sight) while still allowing cooperative strategies to emerge.

D. Catch radius & Team reward

A team capture is defined as a scenario in which both predators are within a specified radius of the prey at the time of capture. In this experiment, we investigate how variations in the team reward and catch radius influence key performance metrics. Figure 2a presents the lone-wolf capture rate across various catch radius and team reward configurations. Contrary to the findings in the 2D grid-based setting by Leibo et al. [18], our results reveal a counterintuitive trend: the rate of lone captures appears to decrease as both the catch radius and the team reward decrease. We hypothesize that this decrease does not reflect true collaborative behavior, but rather an emergent adaptation to environmental constraints. Specifically, with limited incentive for team captures and a smaller effective radius, agents may independently intensify their pursuit of the prey, resulting in more coincidental group captures. However, as shown in Figure 2b, the proximity rate between predators increases under configurations that promote cooperation. This suggests that higher team rewards encourage agents to remain closer to one another, indicating a tendency toward more collaborative behavior.



(a) (b)

Figure 2: (a) Lone-wolf capture rate and (b) predator proximity rate, each as a function of catch radius and team reward.

E. Vision parameters

We aimed to evaluate the influence of visual perception on the hunting strategies of predators by conducting multiple experiments on key vision parameters of ray-casting, specifically the field of view (vision angle) and ray length. We observe that, for a fixed field of view (FOV), increasing the vision distance leads to a higher lone-wolf capture rate. When both the FOV and vision range are limited, predators receive less environmental information and exhibit a more cautious waiting strategy, as illustrated in video [19]. Conversely, as the visual spectrum broadens, a behavior similar to that discussed in the previous section emerges where predators tend to rush toward the prey independently, without coordinating with their peer, which increases the incidence of lone captures (see video [20]). Interestingly, in configurations with a large vision range (e.g., 250° FOV and 40 units of vision distance), we observe a significant increase in proximity rate between predators. This suggests that when agents are better informed, they may naturally maintain closer positions, potentially facilitating more cooperative interactions (see video [21]).

F. Memory

For our baseline implementation, we used a training setup with 10 stacked observation vectors, effectively buffering past observations to give agents limited temporal context.

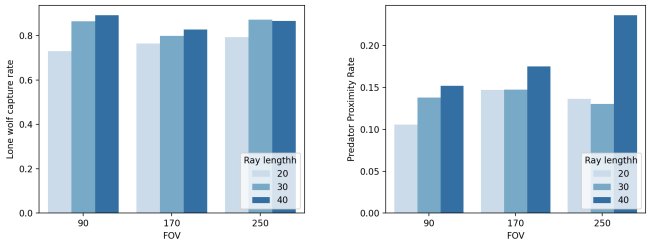


Figure 3: Left : lone-wolf capture rate, Right : predator proximity rate, with respect to the FOV and ray length

This approach allows predators to infer quantities such as velocity and acceleration from sequential frames. However, we aimed to investigate whether more sophisticated memory mechanisms could facilitate deeper cooperation. To this end, we incorporated a Long Short-Term Memory (LSTM) [22] network which provides an internal memory state that can persist over longer time horizons, allowing agents to retain relevant information beyond the short buffer of stacked vectors. To evaluate the impact of the LSTM, we varied two key parameters: the sequence length, which determines how many past time steps are included in each input, and the memory size, which defines the dimensionality of the hidden state of the LSTM. We observed that using an LSTM led to a broader spatial distribution of prey capture. Predators were able to catch prey across the entire map, whereas in the baseline setup, captures were mostly confined to the corners. This indicates that the LSTM helps agents remember where the prey was seen, enabling them to continue chasing it even when it is no longer in direct line of sight due to obstacles.

Increasing the memory size from 64 to 128 reinforced this effect (see Figure 4), as the larger internal state allows for better retention of relevant past observations. We also found that increasing the sequence length led to more varied capture positions with a memory size of 64, but had limited additional impact at 128 possibly because the larger memory already stores sufficient temporal context. This effect may also depend on the map size; on larger maps, even with memory size 128, longer sequences could still provide added benefits.

Interestingly, the rate of lone wolf captures did not significantly increase or decrease with the LSTM (see Figure 5). This suggests that the agents do not use their internal memory to encode or track the position of their teammate. In other words, the hidden state doesn't seem to capture whether another wolf is nearby or coordinate actions based on that information. Similarly, the overall capture rate did not improve with the use of LSTM (also shown in Figure 5). This is somewhat surprising given that predators can now catch prey in more locations across the map, but over 90% of prey are already being caught without LSTM, which makes further improvements inherently difficult. However, when looking at the average number of steps prey survived, we see a clear improvement: without LSTM, the average survival time was 3697.744, while the LSTM-based setups yielded an average of 2854.923, representing a 22.79% increase in capture speed. This suggests that although the total number of captures may not increase, prey are being caught more quickly.

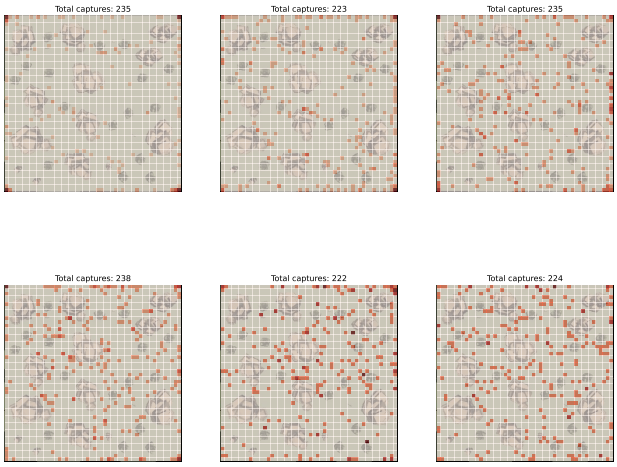


Figure 4: Position of captures with LSTM memory (rows = memory sizes 64/128; columns = sequence lengths 16, 32, 64).

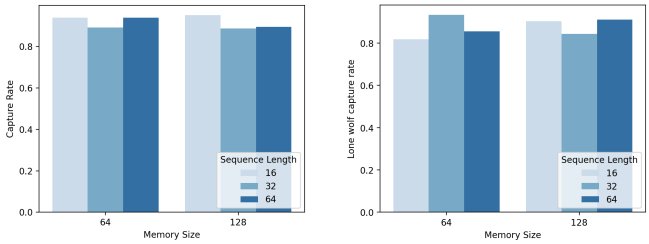


Figure 5: Left : capture rate, Right : Lone-wolf capture rate, with respect to the LSTM parameters (memory size, sequence length)

G. Smelling modality

Although predators possess a wide field of view, they cannot perceive what is directly behind them, which can lead to missing prey that is very close but just outside their field of view. Wolves possess highly developed olfactory senses, which they use to track prey and other wolves [14]. To approximate this ability, we introduced an additional sensory modality aimed at simulating smell and assessing its effect on emergent strategies.

We initially implemented the smelling modality using a discretized scent map with a trace-based system, where agents left behind a fading scent that decayed over time and distance. Additionally, we explored a compass-like mechanism that indicated the direction of the prey and the

other predator when within a predefined detection radius. However, both methods produced poor results, appearing to confuse the agents rather than facilitate improved coordination or collaboration.

H. Leaving the SSD framework

Despite extensive tuning of positive solo and team rewards within the SSD paradigm, agents persistently adopted "lone-wolf" strategies: as soon as one predator spotted the prey, it broke formation and gave chase, even when a cooperative capture would have yielded higher joint payoff [20]. To overcome this failure to achieve genuine cooperation under the SSD reward structure, we experimented with *negative solo rewards*, effectively penalizing individual captures rather than merely rewarding team hunts.

- **Solo capture penalty** ($r_{\text{solo}} < 0$): We set the base solo reward to a negative value ($r_{\text{solo}} = -1.0$ or -10.0), so that any single-wolf capture results in a net penalty.
- **Other penalties and rewards**: To maintain incentive for cooperation, team captures remained positively scaled. All other penalties (time step, collisions) and auxiliary rewards (vision) were kept as in the baseline.

We observe that predators successfully locate one another at the start of each episode and remain in close proximity (average predator distance along the episode going from an average of 35 for $r_{solo} = 1$ to 26 for -1 and 25 for -10). However, as soon as one predator visually detects the prey, it immediately breaks formation and pursues the prey alone, abandoning its partner. We also note that the overall number of captures decreases as we add more penalty (from 140 to 49 over 250 episodes). Regarding the lone-wolf rate, we don't observe a decrease as hoped as explained by the observed behavior above ($L_{r_{solo}=-1} = 0.8$ and $L_{r_{solo}=-10} = 0.85$). But that is also due to the decrease in number of captures. We hypothesize that going out of the SSD might require more training as you would need to learn a 2 steps strategy: first successfully locating the other wolf and then staying in close proximity up until finding the prey together and attacking without leaving the other one behind. Further work could also introduce a small bonus for maintaining close proximity over time.

V. Conclusion & Limitations

A. Limitations

Our approach, while effective in modeling sequential social dilemmas in a continuous 3D setting, exhibits several constraints. The strong interdependence of hyperparameters (e.g., learning rates, reward scales, vision-reward coefficients) can yield non-linear shifts in behavior and complicate reproducibility. Furthermore, our summary metrics, such as capture rates and inter-agent distances, reduce rich coordination dynamics to thresholded outcomes, potentially overlooking subtle cooperative strategies. The use of reward shaping and a fixed capture radius, though practical, embeds inductive biases that may not generalize without substantial retuning. Lastly, occasional oscillations in policy performance suggest that longer training schedules or stabilization techniques (e.g., curriculum learning) may be necessary for robust convergence. In the base paper, the authors conduct training over 40M steps within a significantly simpler environment. Replicating this level of training proved challenging in our case due to limited computational resources.

B. Conclusion

In this project, we extended the Wolfpack game to a realistic 3D continuous environment to investigate how increased environmental complexity influences cooperation in sequential social dilemmas. Our results show that while certain settings encourage closer proximity between agents, traditional SSD reward structures often fall short in fostering stable cooperative behavior. Although some modifications reduced inter-agent distances, they did not significantly lower lone-wolf capture rates, indicating limited real cooperation. These findings highlight the need for more sophisticated design choices and richer perceptual inputs. Overall, our results underscore that increasing environmental realism can both enrich and make more complex the dynamics of cooperation in multi-agent systems. While more informative observations and memory lead to interesting strategic possibilities, they do not automatically guarantee cooperative strategies. Future work should explore adaptive or curriculum-based incentive schemes, dynamic partner-reward sharing, and scaling to larger teams to further bridge the gap between theoretical SSD and real-world multi-agent coordination.

VI. Individual contributions

A.M. and M.L. established the Unity-based infrastructure to support training and inference in a MARL framework. A.M. and J.S. developed the core C# training logic and implemented the Wolfpack game, M.L. and J.S. integrated additional sensory modalities into agent observations and T.S. developped the prey heuristic. T.S. conducted multiple experiments especially with the vision reward. We all designed, executed and analyzed the experiments.

REFERENCES

[1] J. Z. Leibo, V. Zambaldi, M. Lanctot, J. Marecki, and T. Graepel, "Multi-agent reinforcement learning in sequential social dilemmas," *arXiv preprint arXiv:1702.03037*, 2017.

[2] E. Hughes, J. Z. Leibo, M. G. Phillips, K. Tuyls, E. A. Duéñez-Guzmán, A. G. Castañeda, I. Dunning, T. Zhu, K. R. McKee, R. Koster, H. Roff, and T. Graepel, "Inequity aversion improves cooperation in intertemporal social dilemmas," 2018. [Online]. Available: https://arxiv.org/abs/1803.08884

[3] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: a survey," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 895–943, 2022. [Online]. Available: https://doi.org/10.1007/s10462-021-09996-w

[4] M. Tan, "Multi-agent reinforcement learning: independent versus cooperative agents," in *Proceedings of the Tenth International Conference on International Conference on Machine Learning*, ser. ICML'93. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, p. 330–337.

[5] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. N. Foerster, and S. Whiteson, "QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning," *CoRR*, vol. abs/1803.11485, 2018. [Online]. Available: http://arxiv.org/abs/1803.11485

[6] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[7] C. Beattie, J. Z. Leibo, D. Teplyashin, T. Ward, M. Wainwright, H. Küttler, A. Lefrancq, S. Green, V. Valdés, A. Sadik *et al.*, "Deepmind lab," *arXiv preprint arXiv:1612.03801*, 2016.

[8] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaśkowski, "Vizdoom: A doom-based ai research platform for visual reinforcement learning," in *2016 IEEE conference on computational intelligence and games (CIG)*. IEEE, 2016, pp. 1–8.

[9] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar *et al.*, "Unity: A general platform for intelligent agents," *arXiv preprint arXiv:1809.02627*, 2018.

[10] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[11] M. J. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps." in *AAAI fall symposia*, vol. 45, 2015, p. 141.

[12] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[13] M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan *et al.*, "Population based training of neural networks," *arXiv preprint arXiv:1711.09846*, 2017.

[14] P. Frame, "Wolves: Behavior, ecology, and conservation, edited by l. david mech and luigi boitani," *ARCTIC*, vol. 57, 01 2004.

[15] M. Le Bras, A. Miele, J. Siffert, and T. Stanic, "Wolfpack - random prey," https://youtu.be/pTMpKbEIfx4, 2025.

[16] ——, "Wolfpack - smart prey," https://youtu.be/PXQMMb0BEso, 2025.

[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[18] J. Z. Leibo, E. A. Dueñez-Guzman, A. Vezhnevets, J. P. Agapiou, P. Sunehag, R. Koster, J. Matyas, C. Beattie, I. Mordatch, and T. Graepel, "Scalable evaluation of multi-agent reinforcement learning with melting pot," in *International conference on machine learning*. PMLR, 2021, pp. 6187–6199.

[19] M. Le Bras, A. Miele, J. Siffert, and T. Stanic, "Wolfpack - chase solo, catch group," https://youtu.be/sQnVzN31HUE, 2025.

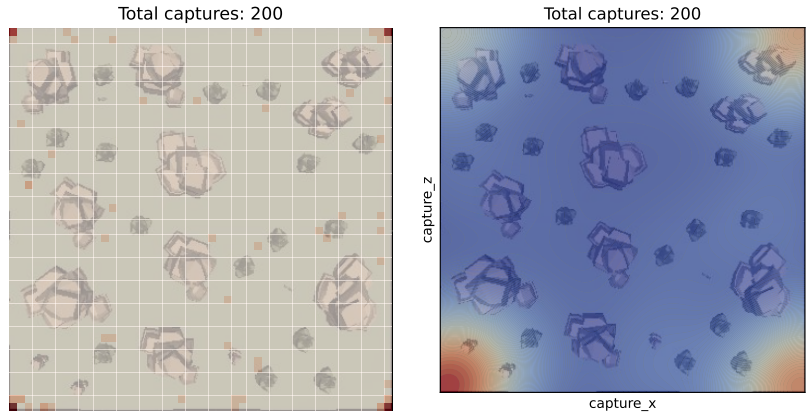
[20] ——, "Wolfpack - stay close then break formation," https://youtu.be/bLVDm7BLcVc, 2025.

[21] ——, "Wolfpack - stay close," https://youtu.be/oddvULAqyYk, 2025.

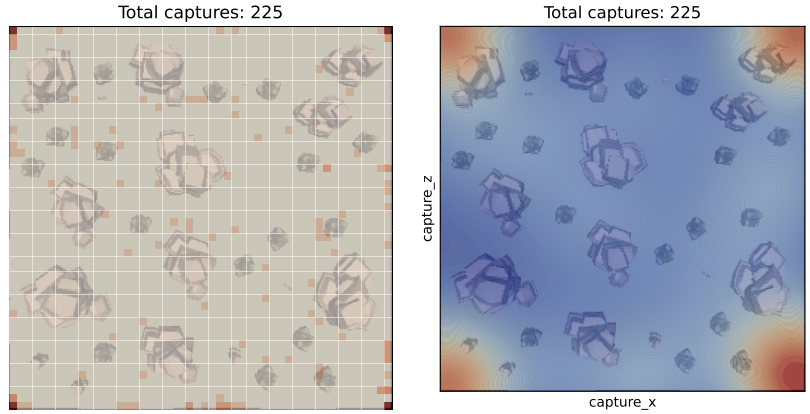
[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, Nov. 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

APPENDIX

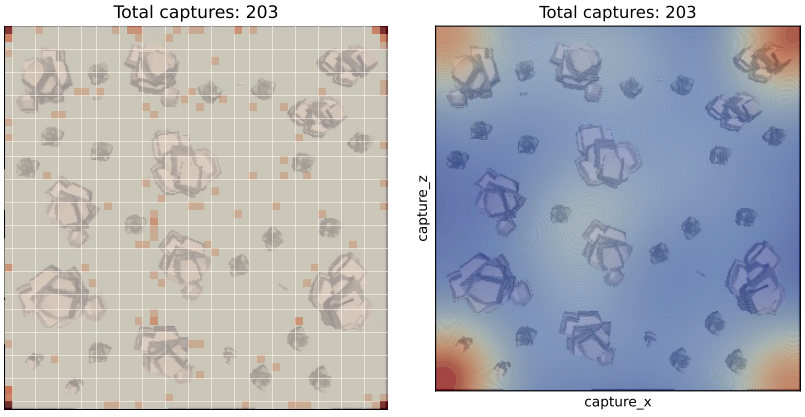
A. Vision variations



(a) 45° FOV, discrete (b) 45° FOV, continuous

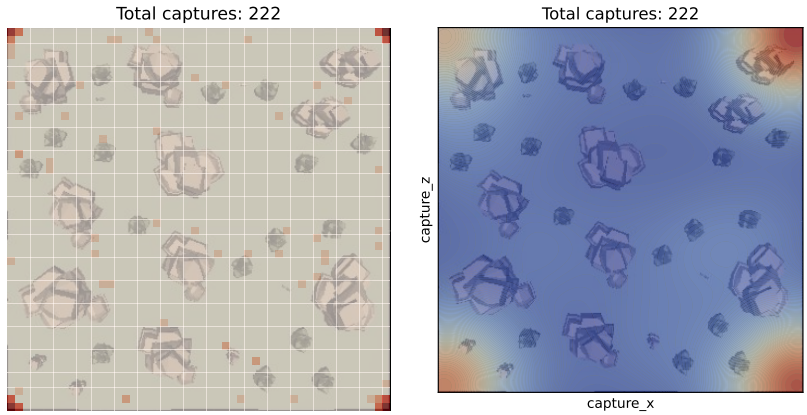


(c) 85° FOV, discrete (d) 85° FOV, continuous

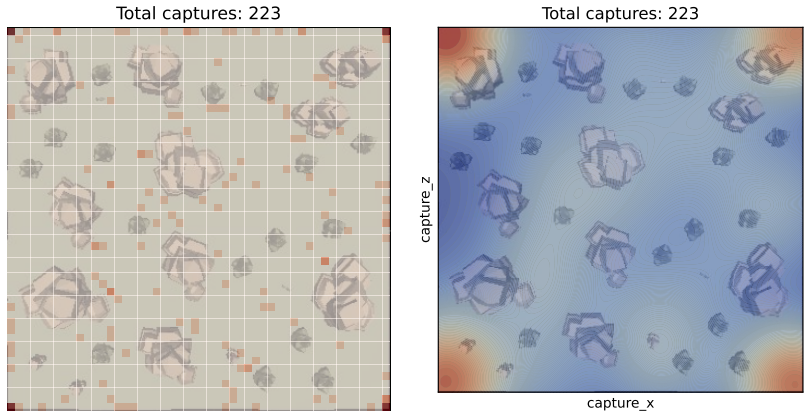


(e) 125° FOV, discrete (f) 125° FOV, continuous

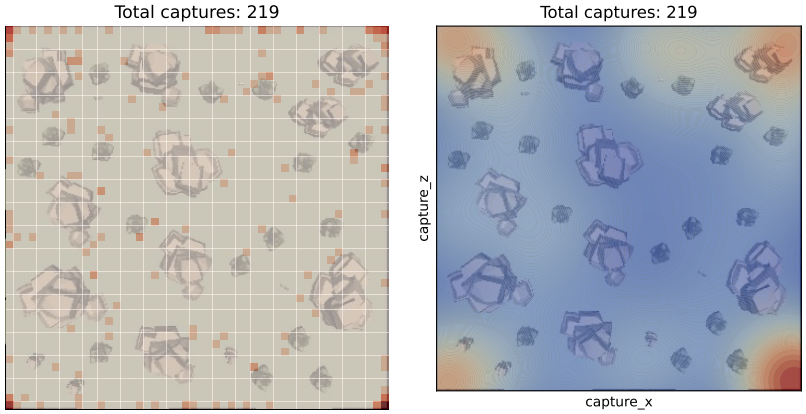
Figure 6: Capture-position heatmaps for ray length = 20 (discrete vs. continuous).



(a) 45° FOV, discrete (b) 45° FOV, continuous

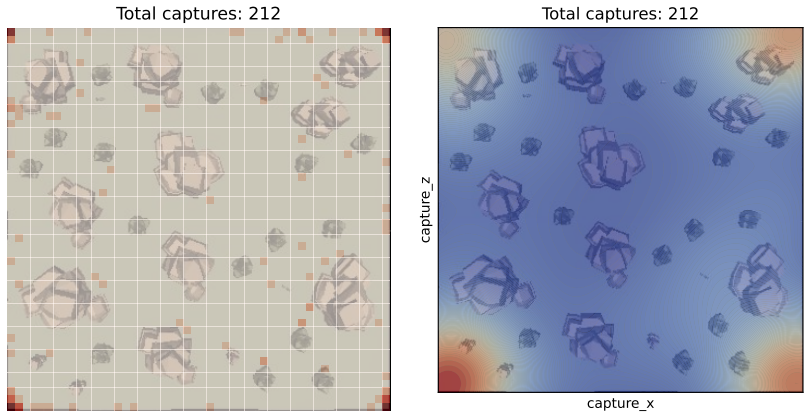


(c) 85° FOV, discrete (d) 85° FOV, continuous

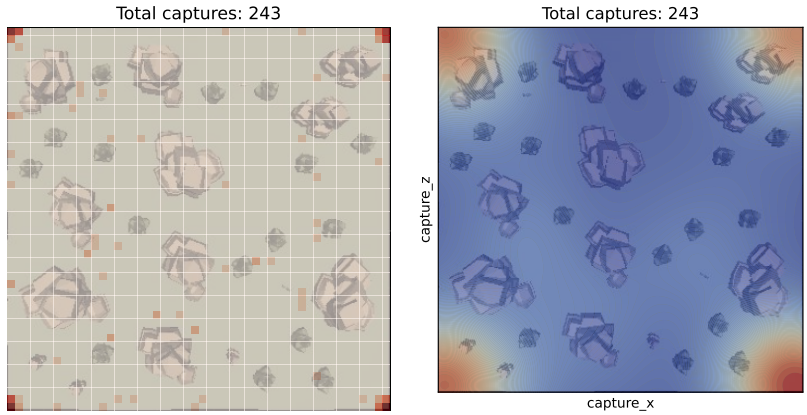


(e) 125° FOV, discrete (f) 125° FOV, continuous

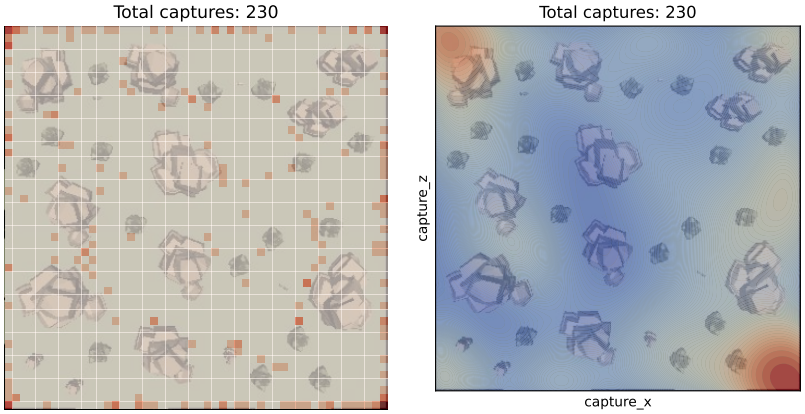
Figure 7: Capture-position heatmaps for ray length = 30 (discrete vs. continuous).



(a) 45° FOV, discrete (b) 45° FOV, continuous

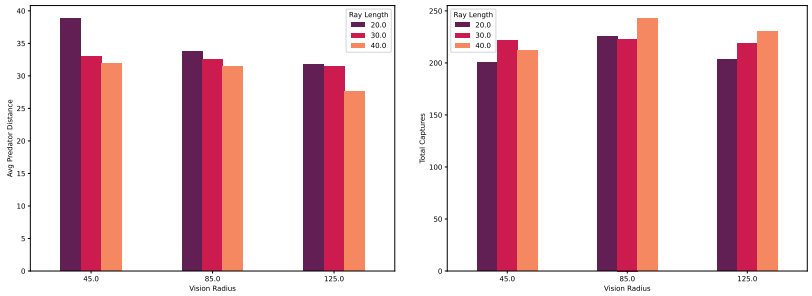


(c) 85° FOV, discrete (d) 85° FOV, continuous

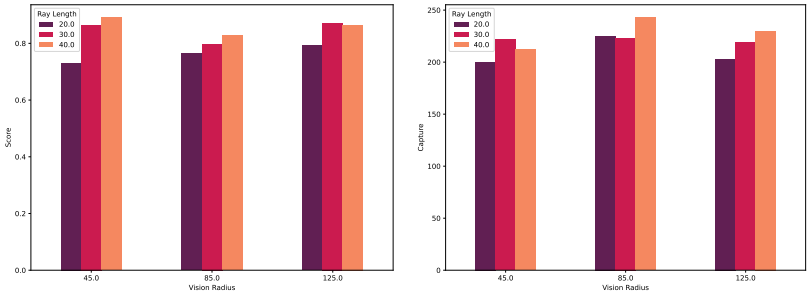


(e) 125° FOV, discrete (f) 125° FOV, continuous

Figure 8: Capture-position heatmaps for ray length = 40 (discrete vs. continuous).

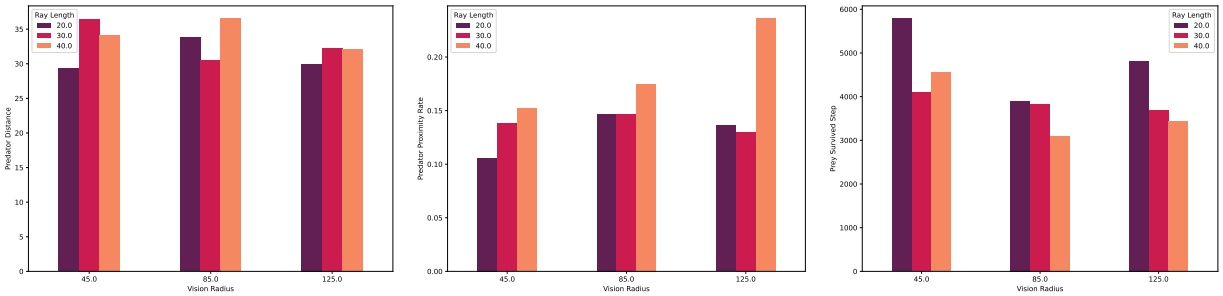


(a) Average predator distance (b) Total captures



(c) Lone wolf capture rate (d) Capture rate

Figure 9: Grouped metrics (2×2): average predator distance, total captures, score, capture rate.



(a) Predator distance (b) Predator proximity rate (c) Prey-survival steps

Figure 10: Additional grouped metrics (1×3): predator distance, predator proximity rate, prey-survival steps.

B. LSTM

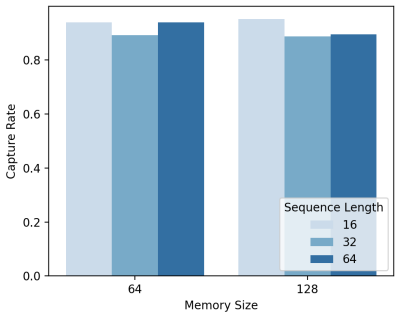
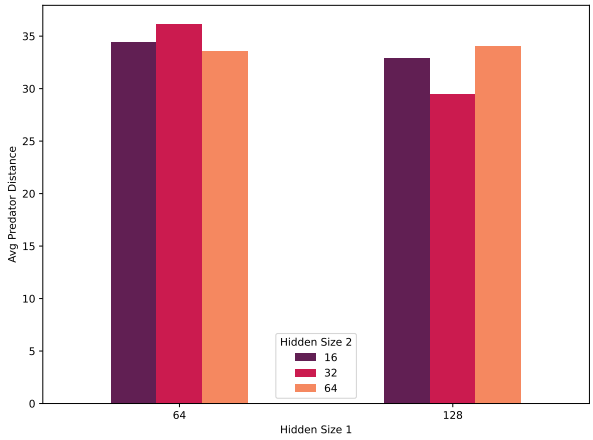
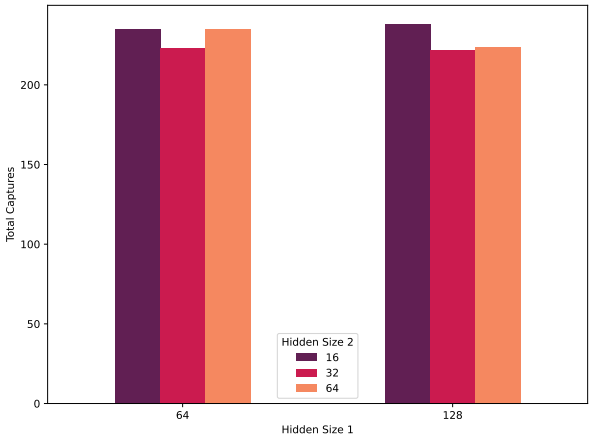


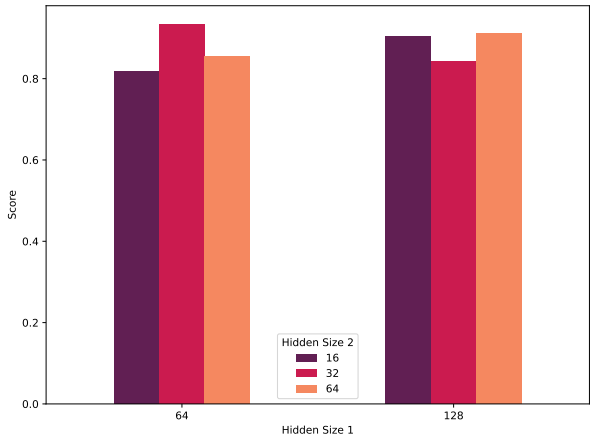
Figure 11: LSTM capture-rate comparison.



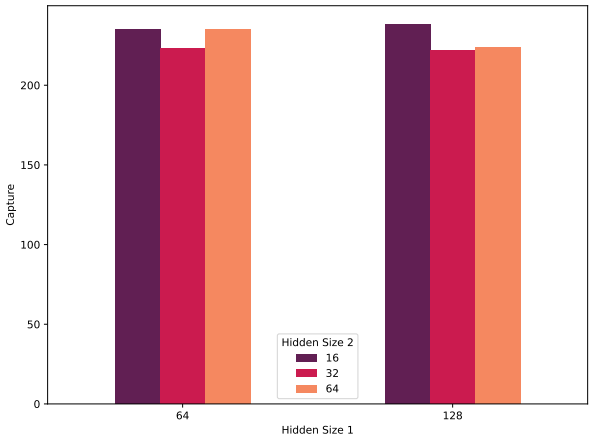
(a) Average predator distance



(b) Total captures

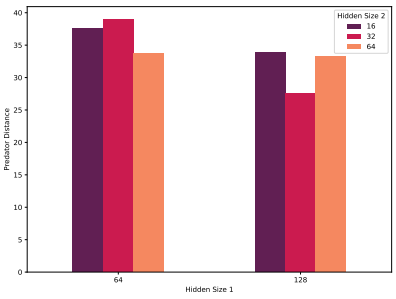


(c) Lone wolf capture rate

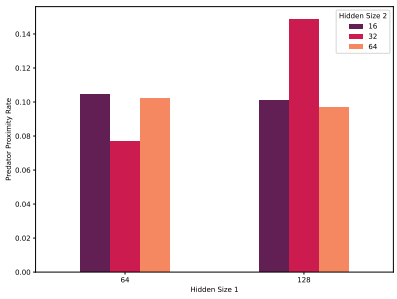


(d) Capture rate

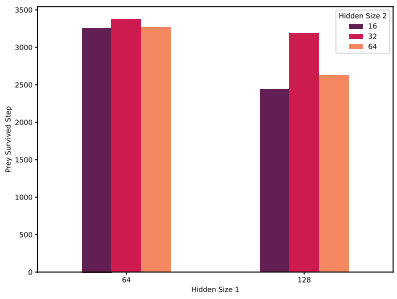
Figure 12: Grouped metrics (2×2 layout): average predator distance, total captures, score, capture rate.



(a) Predator distance



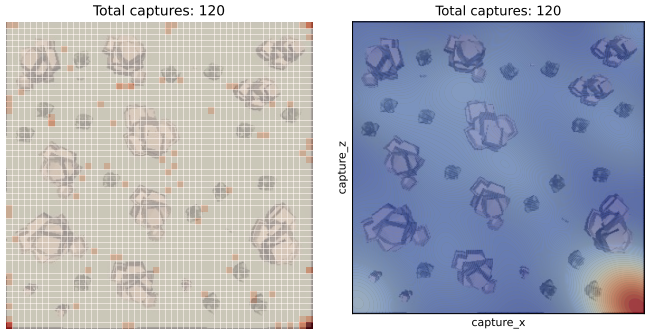
(b) Predator proximity rate



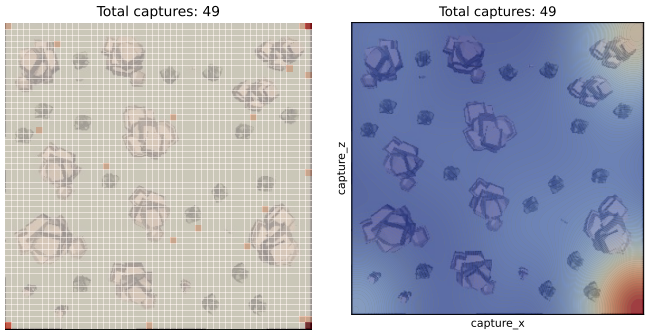
(c) Prey-survival steps

Figure 13: Additional grouped metrics (1×3 layout): predator distance, predator proximity rate, prey-survival steps.

C. Negative solo reward

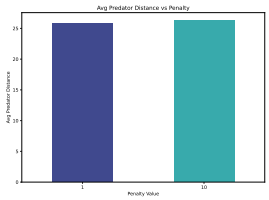


(a) $r_{\text{solo}} = -1$ (discrete) (b) $r_{\text{solo}} = -1$ (continuous)

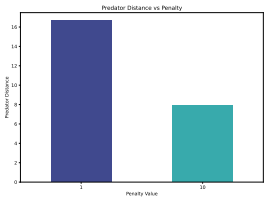


(c) $r_{\text{solo}} = -10$ (discrete) (d) $r_{\text{solo}} = -10$ (continuous)

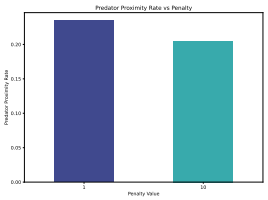
Figure 14: Effect of negative solo-reward penalties on cooperation under discrete vs. continuous vision.



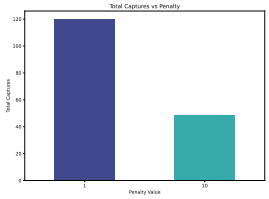
(a) Average predator dist.



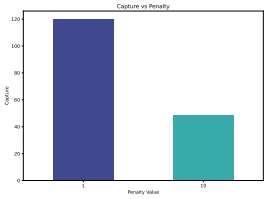
(b) Predator distance



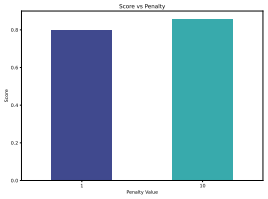
(c) Predator proximity rate



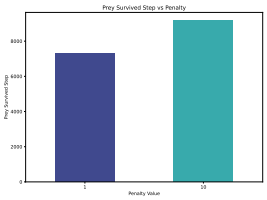
(d) Total captures



(e) Capture rate



(f) Score



(g) Prey-survival steps

Figure 15: Summary of spatial and performance metrics under negative solo-reward penalties.