# Learning Drone Navigation in Caves Using Deep Reinforcement Learning The Trade-offs of LiDAR Sensors

**Tevlen Naidoo (2429493)**
2429493@students.wits.ac.za
School of Computer Science and Applied Mathematics
University of the Witwatersrand

## Abstract

This study investigates the relationship between LiDAR sensor configurations and learning efficiency in deep reinforcement learning for autonomous drone navigation in cave environments. Using AirSim simulation and Proximal Policy Optimisation (PPO), we systematically evaluated four key LiDAR parameters: number of channels, range, points per second, and rotations per second, each tested across low, standard, and high-quality configurations. Our findings reveal a consistent pattern of diminishing returns, where improvements beyond standard configurations yield minimal performance benefits. Standard configurations demonstrated comparable performance to high-quality settings. Additional experiments with real-world LiDAR sensor specifications confirmed these findings, suggesting that mid-range sensors offer optimal cost-effectiveness for cave navigation applications. This research provides practical insights for robotics engineers selecting LiDAR configurations for autonomous cave navigation systems, demonstrating that expensive, high-end sensors may not necessarily translate to proportionally better learning outcomes.

## Introduction

Drones have become essential tools across various fields, enabling tasks such as aerial photography (Cheng, 2015), environmental monitoring (Rohi, Ofualagba, and others, 2020; Gallacher, 2016), and infrastructure inspection (Besada et al., 2018; Seo, Duque, and Wacker, 2018). The development of autonomous navigation capabilities for drones represents a crucial advancement in robotics and artificial intelligence, with reinforcement learning emerging as a promising approach for teaching drones to navigate complex environments.

To enhance their autonomy, drones often employ advanced sensors, with LiDAR (Light Detection and Ranging) being a key technology. LiDAR helps drones create detailed 3D maps of their surroundings, which is vital for effective navigation in intricate cave systems where GPS is ineffective and navigating complex terrain is crucial. However, the integration of sensors poses challenges; issues like noise, limited range, and varying resolutions can impact performance and learning outcomes.



To date, there has been no comprehensive study investigating how different LiDAR sensor configurations affect a drone's ability to learn autonomous navigation behaviours. This study addresses this gap by examining how various LiDAR configurations affect the drone's learning process when navigating caves. Specifically, we aim to identify at what point enhancements in sensor quality lead to diminishing returns in the drone's ability to learn autonomously. By exploring this tradeoff, we hope to provide insights that can improve the deployment of sensing technologies in autonomous systems.

Our results revealed a consistent pattern of diminishing returns across all LiDAR parameters tested. Notably, high-quality sensor configurations showed minimal performance improvements over standard configurations, despite significant increases in sensor specifications. These findings were further validated through experiments using real-world LiDAR sensor parameters, suggesting that mid-range sensors offer the optimal balance between cost and performance for autonomous cave navigation.

Following this introduction, 'Background' presents the background necessary to understand the context and significance of the research. 'Related Works' reviews related work in the field, highlighting key advancements and methodologies. In 'Methodology', we detail our approach, including experimental setup and sensor configurations. The 'Results' section outlines the results of our experiments, followed by a discussion that interprets these findings. Here, we also discuss the limitations and suggest directions for future research. Finally, we conclude the article by summarising the key insights and expanding on the contributions of this research.

## Background

This section provides an overview of key concepts underpinning this research. We begin with Unmanned Aerial Vehicles (UAVs) and their diverse applications. We discuss LiDAR technology, essential for UAV navigation in challenging environments like caves, followed by reinforcement learning (RL) and deep learning, which enable machines to analyse data and learn optimal behaviours through environmental interactions. We then examine deep reinforcement learning (DRL) and Proximal Policy Optimisation (PPO), which are crucial for handling complex state spaces in autonomous navigation.

## Unmanned Aerial Vehicle (UAV)

An Unmanned Aerial Vehicle (UAV)—commonly known as a 'drone'—is defined as a powerful, aerial vehicle that does not carry a human operator, uses aerodynamic forces to provide vehicle lift, can fly autonomously or be piloted remotely, and can be expendable or recoverable. Modern UAVs typically employ a combination of sensors for navigation and control, including GPS for global positioning, Inertial Measurement Units (IMU) for attitude and acceleration data, and various environmental sensors like cameras, LiDAR, or ultrasonic sensors for obstacle detection and mapping. The control architecture usually consists of a low-level flight controller that handles basic stabilisation and motor control and a high-level computer that manages autonomous navigation and mission planning.

UAVs can be operated in several modes. The most basic is manual control, where a human pilot uses a radio transmitter to directly control the drone's movements. More advanced operations use semi-autonomous modes where the drone maintains its stability and follows high-level commands like waypoint navigation. Full autonomy requires the drone to make its own decisions about path planning and obstacle avoidance, typically implemented through algorithms such as SLAM (Simultaneous Localisation and Mapping) for environment mapping, A* or RRT (Rapidly-exploring Random Trees) for path planning, and increasingly, deep learning approaches for perception and decision-making.

Common applications of UAVs include aerial photography, infrastructure inspection, package delivery, and search and rescue operations. Each application demands different control strategies and sensor configurations. For instance, infrastructure inspection might prioritise precise position control and high-resolution imaging, while autonomous navigation in GPS-denied environments like caves or indoor spaces relies more heavily on local sensors and real-time obstacle avoidance algorithms.

**LiDAR** Light Detection and Ranging (LiDAR) is a remote sensing technology that measures distance by emitting laser pulses and recording the time it takes for the pulses to return after reflecting off objects (Collis, 1970). This process enables LiDAR sensors to create detailed 3D point clouds of the environment.

In this study, we are using LiDAR to provide accurate 3D perception of the environment, enabling our autonomous drone to navigate safely and avoid obstacles.

## Reinforcement Learning (RL)

Reinforcement Learning (RL) is a branch of artificial intelligence that focuses on training agents to make optimal decisions in an environment to maximise cumulative reward (Wiering and Van Otterlo, 2012). Unlike supervised learning, which relies on labelled data, RL agents learn through trial and error, interacting with the environment and receiving feedback in the form of rewards or punishments.

The RL problem is formally described as a Markov Decision Process (MDP), defined by the tuple $(S, A, P, R, \gamma)$, where:

- $S$ is the set of all possible states
- $A$ is the set of all possible actions
- $P(s'|s, a)$ is the transition probability function, describing the probability of transitioning to state $s'$ when taking action $a$ in state $s$
- $R(s, a, s')$ is the reward function, defining the immediate reward received when transitioning from state $s$ to $s'$ via action $a$
- $\gamma \in [0, 1]$ is the discount factor that determines the importance of future rewards

**Key Components** The RL framework consists of several essential components that interact continuously:

- **Agent:** The decision-maker interacts with the environment, implementing a policy to select actions.
- **Environment:** The external world that the agent interacts with, is defined by the state space and transition dynamics.
- **State:** The current situation of the agent within the environment, containing all relevant information for decision-making.
- **Action:** The choices available to the agent at each state, are defined by the action space.
- **Reward:** A numerical value that provides immediate feedback on the quality of an action in a given state.

The agent-environment interaction occurs in discrete time steps. At each step $t$, the agent receives the current state $s_t$ and selects an action $a_t$ according to its policy. The environment then transitions to a new state $s_{t+1}$ according to the transition probabilities and provides a reward $r_t$. This process continues until a terminal state is reached or the episode ends.

## Deep Learning

Deep learning is a subfield of machine learning that focuses on training artificial neural networks with multiple layers to learn complex patterns from data. These neural networks are inspired by the structure and function of the human brain, with interconnected nodes (neurons) that process information in a hierarchical manner (LeCun, Bengio, and Hinton, 2015).[1] Neural networks are useful in scenarios where the state space is too large for traditional methods to be practical. In these cases, function approximation becomes necessary, and neural networks serve as powerful universal function approximators, capable of learning and representing complex mappings between inputs and outputs without requiring explicit programming of the underlying relationships.

When dealing with high-dimensional state spaces, such as those encountered in robotic control or image processing, traditional lookup table methods become intractable due to the curse of dimensionality. Neural networks overcome this limitation by learning to approximate the necessary functions through training on examples, generalising to unseen situations, and automatically extracting relevant features from raw input data. This capability makes them particularly well-suited for complex tasks like autonomous navigation, where

[1] https://www.knowledgehut.com/blog/data-science/machine-learning-vs-deep-learning#what-is-machine-learning?

the relationship between sensor inputs and optimal actions cannot be easily specified by hand.

### Deep Reinforcement Learning (DRL)

Deep Reinforcement Learning (DRL) combines the power of deep learning with reinforcement learning to tackle complex tasks that were previously challenging for traditional RL methods (Li, 2017).

By leveraging deep neural networks, DRL agents can learn from raw sensory data—in our case, point cloud data from a LiDAR sensor—and make decisions in high-dimensional state spaces.

### Proximal Policy Optimisation (PPO)

Proximal Policy Optimisation (PPO) is a policy gradient algorithm that has gained significant popularity in reinforcement learning. It is designed to address limitations of previous policy gradient methods, such as the need for careful learning rate tuning and the potential for destructively large policy updates that can lead to training instability (Schulman et al., 2017).

The core innovation of PPO is its objective function, which uses a clipped surrogate loss:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{1}$$

where:

- $r_t(\theta)$ is the probability ratio $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$
- $\hat{A}_t$ is the estimated advantage function
- $\epsilon$ is a hyperparameter that controls the clipping range

The clipping term prevents the ratio between the new and old policies from moving too far from 1, effectively constraining the size of policy updates. This mechanism ensures more stable training compared to traditional policy gradient methods.

PPO has demonstrated remarkable success in training robots for various tasks including grasping, manipulation, and navigation, making it particularly suitable for our study of autonomous drone navigation in cave environments. Its combination of sample efficiency, ease of implementation, and reliable performance has made it a standard choice in deep reinforcement learning applications.

## Related Work

Research in autonomous drone navigation for complex environments like caves has explored various approaches, combining different sensor technologies with control and learning algorithms.

### Classical Control Approaches

Several studies have focused on traditional control methods for drone navigation using LiDAR sensors. Petráček et al. (2021) demonstrated LiDAR's effectiveness in low-light cave environments, leveraging its active illumination properties for reliable distance measurements. Some researchers have explored sensor fusion approaches, particularly combining LiDAR with inertial measurement units (IMU) using Kalman filtering techniques for more robust navigation and mapping capabilities (Ariante et al., 2019).

### Learning-Based Navigation

More recent work has shifted towards learning-based approaches, particularly reinforcement learning (RL), for autonomous navigation. Within this domain, several studies have explored the use of RGB-D sensors as the primary input for RL-based navigation systems.

**Vision-Based Methods** Work by Surmann et al. (2020) presents a deep reinforcement learning framework for autonomous navigation of a real mobile robot in unknown indoor environments. The robot's sensory input combines data from a 2D laser scanner and an RGB-D camera, allowing it to perceive the 3D structure of the environment. The system outputs the linear and angular velocities for the robot's navigation, with the goal of reaching a target location while avoiding collisions. To improve the robustness and safety of the learned policies, the authors pre-train the system in a simulation environment before deploying it on the real robot.

Similarly, Friji et al. (2020) proposed an end-to-end car-following framework for autonomous driving using deep reinforcement learning and RGB-D camera input. Their system, based on an improved Deep Q-Network (DQN) algorithm, learns to follow a leader car while also considering other environmental factors, such as pedestrians and road boundaries. By utilising the depth information from the RGB-D frames, the framework is able to make more informed driving decisions, ensuring safe and effective autonomous navigation in complex road scenarios.

These studies demonstrate the advantages of using RGB-D sensors in combination with deep reinforcement learning for autonomous navigation tasks. The depth information provided by the RGB-D cameras allows the systems to better perceive the 3D structure of the environment, enabling more robust obstacle avoidance and goal-directed navigation, which is particularly relevant for our research on drone navigation in challenging cave environments.

Loquercio et al. (2019) developed a deep reinforcement learning framework for drone navigation in complex indoor environments using only onboard cameras. Their approach, called "DroNet" (Loquercio et al., 2018), was able to navigate through corridors and avoid obstacles effectively.

**LiDAR-Based Methods** Several studies have demonstrated the viability of LiDAR-based RL for autonomous navigation. Miera, Szolc, and Kryjak (2023) achieved notable success using PPO with the RPLIDAR A2 in forest environments, achieving 80% success rates in real-world conditions.

## Methodology

This study investigates how different LiDAR sensor configurations affect the learning efficiency of a drone navigating in cave environments using deep reinforcement learning. Our primary goal is to identify potential trade-offs between sensor quality and learning performance, specifically examining whether higher-quality LiDAR configurations necessarily

lead to better or faster learning outcomes. To systematically explore this relationship, we designed a series of experiments varying key LiDAR parameters while maintaining consistent training conditions and evaluation metrics. This section details our experimental setup, the LiDAR configurations tested, and our evaluation methodology.

## Experimental Setup

**Simulation Environment** The experimental framework was implemented using AirSim (Shah et al., 2018), a high-fidelity simulator for drones and autonomous vehicles. A cave environment was selected as the testing ground to evaluate autonomous navigation capabilities under challenging conditions. The environment utilised the 'Soul: Cave' asset[2], which presents complex geometric features and confined spaces, making it an ideal testbed for evaluating LiDAR-based navigation systems, as seen in Figure 1.

Figure 1: Image showing AirSim using the 'Soul: Cave' environment

The state space consists of the normalised LiDAR point cloud data representing the drone's surroundings, combined with the relative position of the goal with respect to the drone's current position. For the action space, the drone operates with three continuous control dimensions, each normalised to the range $[-1, 1]$, corresponding to forward/backwards, left/right, and up/down movements. These normalised values are then mapped to actual velocity commands, with the maximum velocity capped at 1 meter per second to ensure stable and safe navigation within the confined cave environment.

**Deep Reinforcement Learning Framework** The study employed Proximal Policy Optimisation (PPO) as the deep reinforcement learning algorithm, implemented using the `stable-baselines3` (Raffin et al., 2021) library. The simulation environment was wrapped in OpenAI Gym's (Brockman, 2016) interface to standardise the interaction between the environment and the learning algorithm.

**The Reward Function** The primary task is to train a drone to navigate from a fixed starting position to a constant goal position within a cave environment while avoiding collisions with cave walls and obstacles. Each training episode terminates under three conditions: when the drone successfully reaches within 2 meters of the goal position, collides with an obstacle, or takes longer than 200 timesteps.

The reward function was designed to encourage efficient goal-directed navigation while avoiding collisions:

$$r = \begin{cases} 10 & \text{if } ||\vec{p} - \vec{g}||_2 < 2 \\ -10 & \text{if a collision occurs} \\ \vec{v} \cdot \vec{d} & \text{otherwise} \end{cases} \quad (2)$$

where:

- $\vec{p}$ denotes the current position of the drone,
- $\vec{g}$ represents the position of the target goal,
- $\vec{v}$ indicates the velocity vector of the drone, and
- $\vec{d}$ is the unit vector pointing in the direction of the goal.

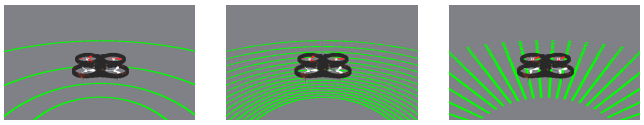
The episode terminates immediately when $||\vec{p} - \vec{g}||_2 < 2$ (success), when a collision occurs (failure); or simply if the drone takes 200 timesteps without making it to the goal or collides. In the latter case, the reward it takes is the latest dot product. During training, successful navigation is rewarded with $+10$, collisions are penalised with $-10$, and intermediate states receive a reward based on the alignment of the drone's velocity vector with the direction to the goal ($\vec{v} \cdot \vec{d}$), encouraging efficient path-taking behaviour.

Convergence was defined as the point at which the model consistently achieved the maximum reward of $+10$, indicating reliable goal-reaching behaviour.

## Experimental Design

**LiDAR Parameter Space** The study investigates the impact of LiDAR sensor characteristics on training efficiency in deep reinforcement learning for autonomous cave navigation. Four key LiDAR parameters were identified and varied across three quality levels (low, standard, and high), resulting in 12 distinct configurations, as can be seen in Table 1.

**NumberOfChannels**: The number of channels represents the vertical resolution of the LiDAR sensor, with higher channel counts providing more detailed spatial information. A visual representation can be seen in Figure 2.



(a) 4 channels (b) 16 channels (c) 64 channels

Figure 2: Images showing the different configurations for `NumberOfChannels`

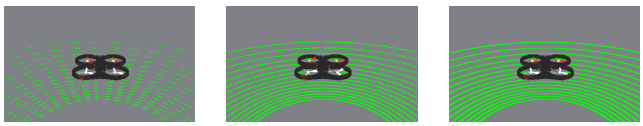
**Range**: The range indicates the maximum effective distance measurement capability, crucial for detecting obstacles and navigation targets in cave environments.

**PointsPerSecond**: Points per second measure the sensor's data capture density, with higher values providing more granular environmental mapping. A visual representation can be seen in Figure 3.

[2] https://www.fab.com/listings/75f42402-40bb-4a1b-b557-18e2c9604273

Table 1: The LiDAR parameters, their descriptions, and the different configurations

| Parameter | Description | Configuration | | |
|---|---|---|---|---|
| | | Low | Standard | High |
| `NumberOfChannels` | Number of channels/lasers of the LiDAR | 4 channels | 16 channels | 64 channels |
| `Range` | Range, in meters | 10 meters | 100 meters | 200 meters |
| `PointsPerSecond` | Number of points captured per second | 20 000 points/$s$ | 100 000 points/$s$ | 500 000 points/$s$ |
| `RotationsPerSecond` | Rotations per second | 5 rotations/$s$ | 30 rotations/$s$ | 60 rotations/$s$ |



(a) 20 000 p/$s$ (b) 100 000 p/$s$ (c) 500 000 p/$s$

Figure 3: Images showing the different configurations for `PointsPerSecond`

**RotationsPerSecond**: Rotations per second determine the angular velocity of the LiDAR sensor, affecting the speed and comprehensiveness of environmental scanning.

## Real-World LiDAR Configuration Analysis

In addition to the systematic parameter exploration, we conducted three supplementary experiments using configurations that approximate real-world LiDAR sensors commonly used in robotics applications. The sensors we used were the RPLIDAR A1 (entry-level)[3], Velodyne VLP-16 (mid-range)[4], and Velodyne HDL-64E (high-end)[5]. These experiments aimed to bridge the gap between our theoretical parameter space and practical implementations, providing insights into how commercially available sensors might perform in cave navigation tasks. Table 2 presents the specifications of the entry-level, mid-range, and high-end LiDAR sensors whose parameters we replicated in our simulation environment.

The configurations were selected to represent different price points and capabilities in the current market, allowing us to evaluate how real-world sensor limitations might impact training outcomes. By including these practical configurations alongside our systematic parameter exploration, we can provide more directly applicable insights for robotics engineers selecting sensors for autonomous cave navigation systems.

These additional experiments followed the same training protocol and evaluation metrics as our primary parameter space exploration, ensuring consistent comparison across all results.

[3] https://www.slamtec.com/en/lidar/a1

[4] https://ouster.com/products/hardware/vlp-16

[5] https://www.mapix.com/wp-content/uploads/2018/07/63-9194_Rev-J_HDL-64E_S3_Spec-Sheet-Web.pdf

**Training Protocol** Each experimental configuration underwent training for 100 000 episodes to evaluate convergence characteristics. The training process began with configuring the AirSim simulation environment and establishing the specified cave layout for consistent testing conditions. Following this initial setup, we adjusted the LiDAR parameters according to our experimental matrix, ensuring each configuration was properly implemented.

The PPO model was then initialised using the default hyperparameters provided by the `stable-baselines3` library. Training execution maintained consistent hyperparameters across all experiments to ensure a fair comparison between different LiDAR configurations. Throughout the training process, we continuously monitored performance metrics and assessed convergence based on the defined reward metrics.

**Performance Metrics** The primary metric for evaluation was the time-to-convergence, measured as the number of episodes required to consistently achieve the maximum reward of $+10$. This indicated that the drone had learned to successfully navigate to the goal while avoiding collisions.

## Data Collection and Analysis

The training data we collected was the episode rewards over time. Based on the episode rewards, we were able to determine the time-to-convergence, and this is what we used to evaluate the training performances.

## Reproducibility Considerations

To ensure reproducibility, the following measures were implemented:

- Consistent initial conditions across all experiments
- Standardised evaluation criteria based on reward thresholds
- Consistent hyperparameters across all training runs

**Environment Configuration** The cave environment was kept constant across all experiments, with:

- Fixed starting position for the drone
- Consistent goal location
- Unchanged cave geometry and dimensions
- Standard lighting conditions

Table 2: Comparison of sensor parameters

| | | Parameters | | | | Pricing |
|---|---|---|---|---|---|---|
| | | NumberOfChannels | Range | PointsPerSecond | RotationsPerSecond | |
| **Sensors** | High-end | 64 | 120.0 | 1 300 000 | 20.0 | ≈R90 000 (≈US$5000) |
| | Mid-range | 16 | 100.0 | 300 000 | 10.0 | ≈R72 000 (≈US$4000) |
| | Entry-level | 1 | 12.0 | 4 000 | 5.5 | ≈R 1 800 (≈US$100) |

## Results

Due to the nature of the reward function, all results were smoothed—for better visualisation and interpretability—using a Running Average smoothing with a window size of 10 000.

All code used to obtain the following results can be found at https://github.com/AwesomeTevv/Honours-Research-Project.

### Parameter-Specific Training Outcomes

This section displays the results of the main experiments.

**NumberOfChannels** The results of training a drone with the specified configurations of the NumberOfChannels parameter can be seen in Figure 4.

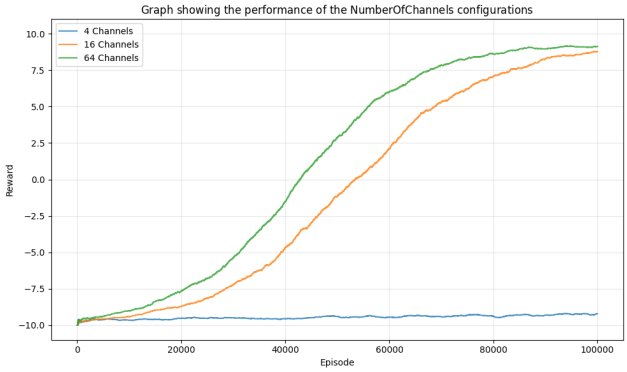



Figure 4: Line graph showing the episode rewards of the drones with differing NumberOfChannels

**Range** The results of training a drone with the specified configurations of the Range parameter can be seen in Figure 5.

**PointsPerSecond** The results of training a drone with the specified configurations of the PointsPerSecond parameter can be seen in Figure 6.

**RotationsPerSecond** The results of training a drone with the specified configurations of the RotationsPerSecond parameter can be seen in Figure 7 on the next page.

### Real-World LiDAR Configuration Performance

The results of training drones using the real-world LiDAR sensor configurations, as seen in Table 2, can be seen in Figure 8 on the next page.

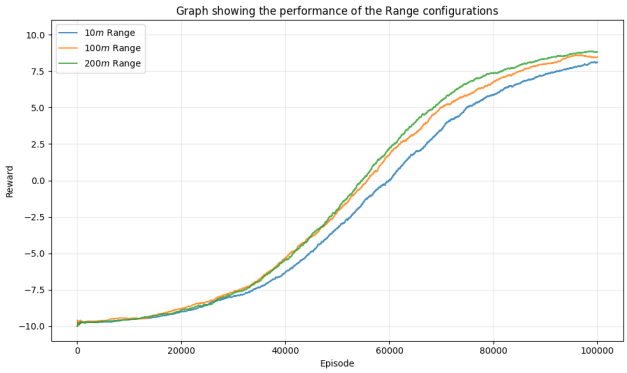



Figure 5: Line graph showing the episode rewards of the drones with differing Range

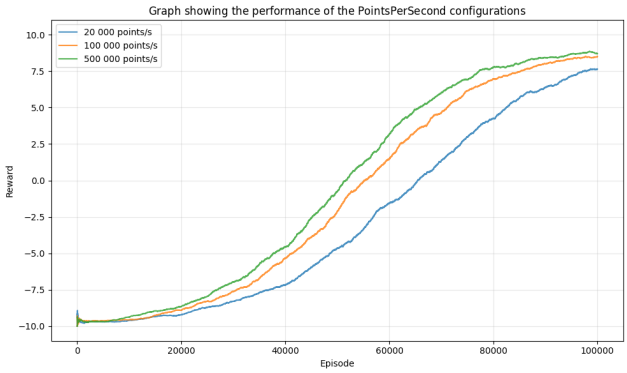



Figure 6: Line graph showing the episode rewards of the drones with differing PointsPerSecond

Furthermore, we visualised the relationship between the price of the sensor and its time-to-convergence. This can be seen in Figure 9 on the following page.

## Discussion

### The Impact of LiDAR Sensor Qualities on Learning

In this section, we discuss the results of the main experimentation.

**`NumberOfChannels`** From Figure 4, we can see that the 4-channel configuration demonstrated significant limitations in learning capability, failing to achieve convergence throughout the training period. This suggests insufficient

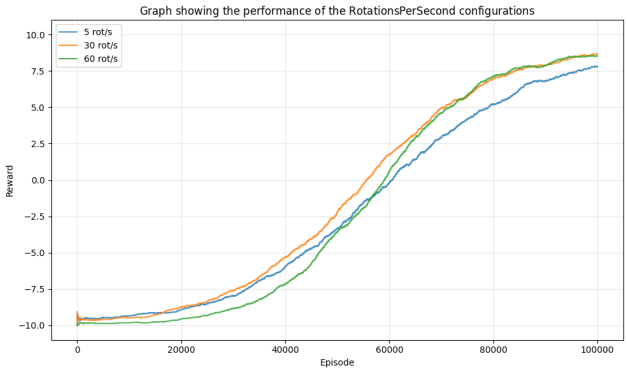



Figure 7: Line graph showing the episode rewards of the drones with differing RotationsPerSecond

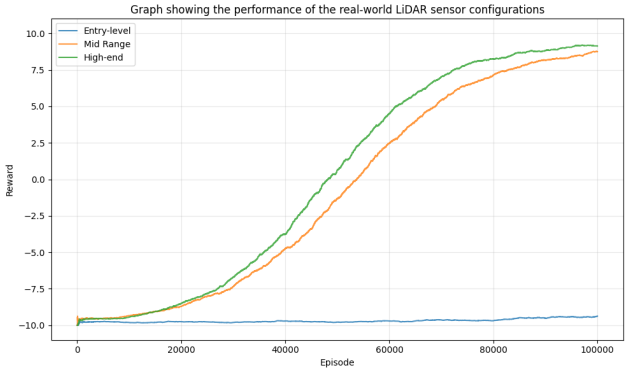



Figure 8: Line graph showing the episode rewards of the drones using the real-world LiDAR configurations

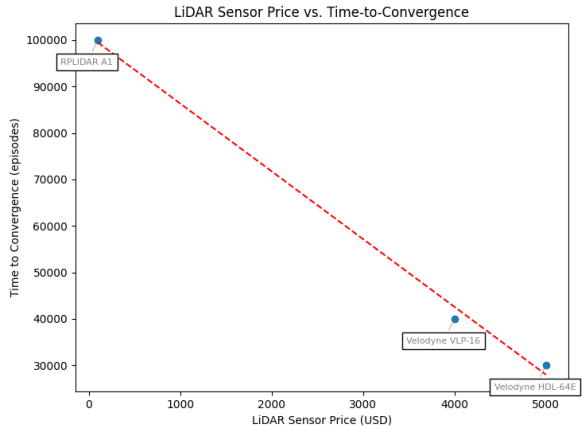



Figure 9: Scatter plot showing the real-world LiDAR sensor prices (in USD) vs. the time-to-convergence

vertical resolution to map the cave environment effectively. In contrast, both 16-channel and 64-channel configurations successfully converged. Notably, the increase from 16 to 64 channels yields the most substantial increase in performance compared to the other parameters, with the 64-channel drone achieving convergence much quicker, and at a faster rate, as indicated by the steeper curve. This suggests that there is merit in opting for a higher-quality sensor if it offers an increase in the number of LiDAR channels.

**`Range`** Interestingly, all range configurations (10-meter, 100-meter, and 200-meter) demonstrated similar learning capabilities, with negligible differences in convergence time and final performance, as can be seen in Figure 5 on the preceding page. The consistent performance across different range settings suggests that this parameter has minimal impact on the drone's ability to learn navigation behaviours in cave environments. This unexpected result might be attributed to the confined nature of cave spaces, where even shorter-range capabilities provide sufficient environmental information for effective learning.

**`PointsPerSecond`** Figure 6 on the previous page shows us that all three configurations—low (20 000 points$/s$), standard (100 000 points$/s$), and high (500 000 points$/s$)—successfully achieved convergence. While the low-density configuration exhibited delayed learning compared to its counterparts, it eventually reached similar performance levels. The standard and high-density configurations showed nearly identical convergence timing and final performance, suggesting that increasing point density beyond the standard configuration offers minimal advantages for training efficiency.

**`RotationsPerSecond`** By looking at Figure 7 we can see that the 60-rotations-per-second model took slightly longer to converge than both the 5- and 30-rotations-per-second models. But ultimately, all three models converged with similar performance metrics. This indicates that while higher rotation speeds can accelerate the learning process, even lower rotation rates can eventually achieve comparable navigation capabilities.

## Comparison of Simulated Configurations with Real-World LiDAR Examples

According to Figure 8, the high-end LiDAR sensor configuration outperforms the mid-range and entry-level configurations throughout the training process. However, the high-end only slightly outperforms the mid-range. This indicates that the higher quality and more expensive high-end sensor enables the drone to learn the navigation task more effectively and achieve better overall performance. But also that after a certain price point, there are diminishing returns.

After visualising the relationship between sensor price and time-to-convergence, in Figure 9, we can see that the trend line shows a clear negative correlation between sensor price and convergence time. This suggests that investing in higher-end, more expensive LiDAR sensors can lead to faster training convergence and potentially better overall navigation performance for the drone. But again, after a certain price point, performances start to be more comparable.

The diminishing returns that can be seen may be the result of the problem we are trying to solve: drone navigation in caves. Perhaps, for a different task, the higher-quality sensors

yield significant improvements even compared to the mid-range sensors.

In summary, the results indicate that the higher quality and more expensive high-end LiDAR sensor configuration enables the drone to learn the navigation task more effectively, achieving better overall performance compared to the mid-range (although marginal) and entry-level configurations. Furthermore, the inverse relationship between sensor price and time-to-convergence suggests that the increased investment in better LiDAR sensors can pay off in terms of faster training and potentially improved navigation capabilities for the drone.

### Comprehensive Analysis and Future Directions

**Trade-offs and Practical Implications** Among all parameters examined, NumberOfChannels emerged as the most critical factor influencing learning effectiveness, being the only parameter where the lowest configuration failed to achieve convergence. This suggests that vertical resolution plays a fundamental role in enabling successful learning for cave navigation tasks, while other parameters primarily affect the speed rather than the ultimate success of learning.

Furthermore, our findings demonstrate a clear pattern of diminishing returns across all LiDAR parameters. While minimum threshold requirements exist for successful learning, exceeding these thresholds with higher-quality sensors often yields minimal additional benefits. This suggests that mid-range LiDAR configurations may offer the optimal balance between cost and performance for cave navigation applications. Organisations implementing drone navigation systems should carefully consider whether the additional cost of high-end sensors is justified by their specific use case.

**Limitations** The following limitations should be considered when interpreting these results. First, our simulation environment, while detailed, may not fully capture all real-world cave complexities such as varying surface reflectivity and atmospheric conditions. Additionally, our experiments focused on a single cave configuration, and results may vary in caves with different geometric characteristics.

**Future Work** Future research should explore sensor fusion approaches that combine LiDAR with complementary technologies, potentially enhancing navigation capabilities while mitigating individual sensor limitations. This could be complemented by the development of adaptive learning algorithms that dynamically adjust to varying sensor capabilities and environmental conditions.

To enhance the generalisability of our findings, future studies should extend experiments across diverse cave geometries and environmental conditions, providing a more comprehensive understanding of how different LiDAR configurations perform across varying scenarios. Additionally, investigating the energy efficiency trade-offs of different sensor configurations would offer valuable insights for practical deployment, particularly in resource-constrained exploration missions.

Finally, real-world validation studies comparing simulated and actual performance metrics would be crucial for bridging the sim-to-real gap and validating the practical applicability of our findings. Such validation would provide essential insights for the deployment of autonomous navigation systems in actual cave exploration missions.

## Conclusion

This research set out to examine how different LiDAR sensor configurations affect a drone's ability to learn autonomous navigation behaviours in cave environments. Through systematic experimentation with four key LiDAR parameters and additional testing of real-world sensor configurations, we have uncovered several significant insights that advance our understanding of sensor-learning relationships in autonomous systems.

Our results consistently demonstrated a clear threshold effect across all parameters, below which learning fails to occur, and above which improvements yield diminishing returns. While low-quality sensors did successfully learn (except for `NumberOfChannels`), the standard- and high-quality specifications routinely outperformed them. However, perhaps most significantly, we found that high-end sensor configurations offered minimal advantages over standard configurations in terms of learning speed and navigation success.

These findings have important implications for the field of autonomous robotics, particularly in the context of cave exploration and navigation. They suggest that the selection of LiDAR sensors for autonomous drone applications should prioritise meeting minimum viable specifications rather than maximising sensor capabilities. This insight could lead to more cost-effective deployment of autonomous navigation systems while maintaining performance standards.

The study also highlighted the complex relationship between sensor capabilities and learning efficiency in deep reinforcement learning systems. While sensor quality is crucial for successful learning, our results indicate that beyond certain thresholds, improvements in sensor specifications do not translate to proportional improvements in learning outcomes. This understanding could inform both the design of future autonomous systems and the development of more efficient training methodologies.

As autonomous drone applications continue to expand, particularly in challenging environments like caves where GPS is unavailable, these insights into the relationship between sensor configurations and learning efficiency will become increasingly valuable. Future work should focus on validating these findings in real-world environments, exploring sensor fusion approaches, and investigating how these relationships might vary across different types of cave geometries and environmental conditions.

This research contributes to the growing body of knowledge in autonomous systems by providing practical guidelines for sensor selection and configuration, while also opening new avenues for investigation into the fundamental relationships between sensor capabilities and learning efficiency in autonomous navigation systems.

## References

Ariante, G.; Papa, U.; Ponte, S.; and Del Core, G. 2019. Uas for positioning and field mapping using lidar and imu sensors data: Kalman filtering and integration. In

*2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 522–527. IEEE.

Besada, J. A.; Bergesio, L.; Campaña, I.; Vaquero-Melchor, D.; López-Araquistain, J.; Bernardos, A. M.; and Casar, J. R. 2018. Drone mission definition and implementation for automated infrastructure inspection using airborne sensors. *Sensors* 18(4):1170.

Brockman, G. 2016. Openai gym. *arXiv preprint arXiv:1606.01540*.

Cheng, E. 2015. *Aerial photography and videography using drones*. Peachpit Press.

Collis, R. 1970. Lidar. *Applied optics* 9(8):1782–1788.

Friji, H.; Ghazzai, H.; Besbes, H.; and Massoud, Y. 2020. A dqn-based autonomous car-following framework using rgbd frames. In *2020 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*, 1–6. IEEE.

Gallacher, D. 2016. Drone applications for environmental management in urban spaces: A review. *International Journal of Sustainable Land Use and Urban Planning* 3(4).

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature* 521(7553):436–444.

Li, Y. 2017. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*.

Loquercio, A.; Maqueda, A. I.; Del-Blanco, C. R.; and Scaramuzza, D. 2018. Dronet: Learning to fly by driving. *IEEE Robotics and Automation Letters* 3(2):1088–1095.

Loquercio, A.; Kaufmann, E.; Ranftl, R.; Dosovitskiy, A.; Koltun, V.; and Scaramuzza, D. 2019. Deep drone racing: From simulation to reality with domain randomization. *IEEE Transactions on Robotics* 36(1):1–14.

Miera, P.; Szolc, H.; and Kryjak, T. 2023. Lidar-based drone navigation with reinforcement learning. *arXiv preprint arXiv:2307.14313*.

Petráček, P.; Krátkỳ, V.; Petrlík, M.; Báča, T.; Kratochvíl, R.; and Saska, M. 2021. Large-scale exploration of cave environments by unmanned aerial vehicles. *IEEE Robotics and Automation Letters* 6(4):7596–7603.

Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; and Dormann, N. 2021. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research* 22(268):1–8.

Rohi, G.; Ofualagba, G.; et al. 2020. Autonomous monitoring, analysis, and countering of air pollution using environmental drones. *Heliyon* 6(1).

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Seo, J.; Duque, L.; and Wacker, J. 2018. Drone-enabled bridge inspection methodology and application. *Automation in construction* 94:112–126.

Shah, S.; Dey, D.; Lovett, C.; and Kapoor, A. 2018. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics: Results of the 11th International Conference*, 621–635. Springer.

Surmann, H.; Jestel, C.; Marchel, R.; Musberg, F.; Elhadj, H.; and Ardani, M. 2020. Deep reinforcement learning for real autonomous mobile robot navigation in indoor environments. *arXiv preprint arXiv:2005.13857*.

Wiering, M. A., and Van Otterlo, M. 2012. Reinforcement learning. *Adaptation, learning, and optimization* 12(3):729.