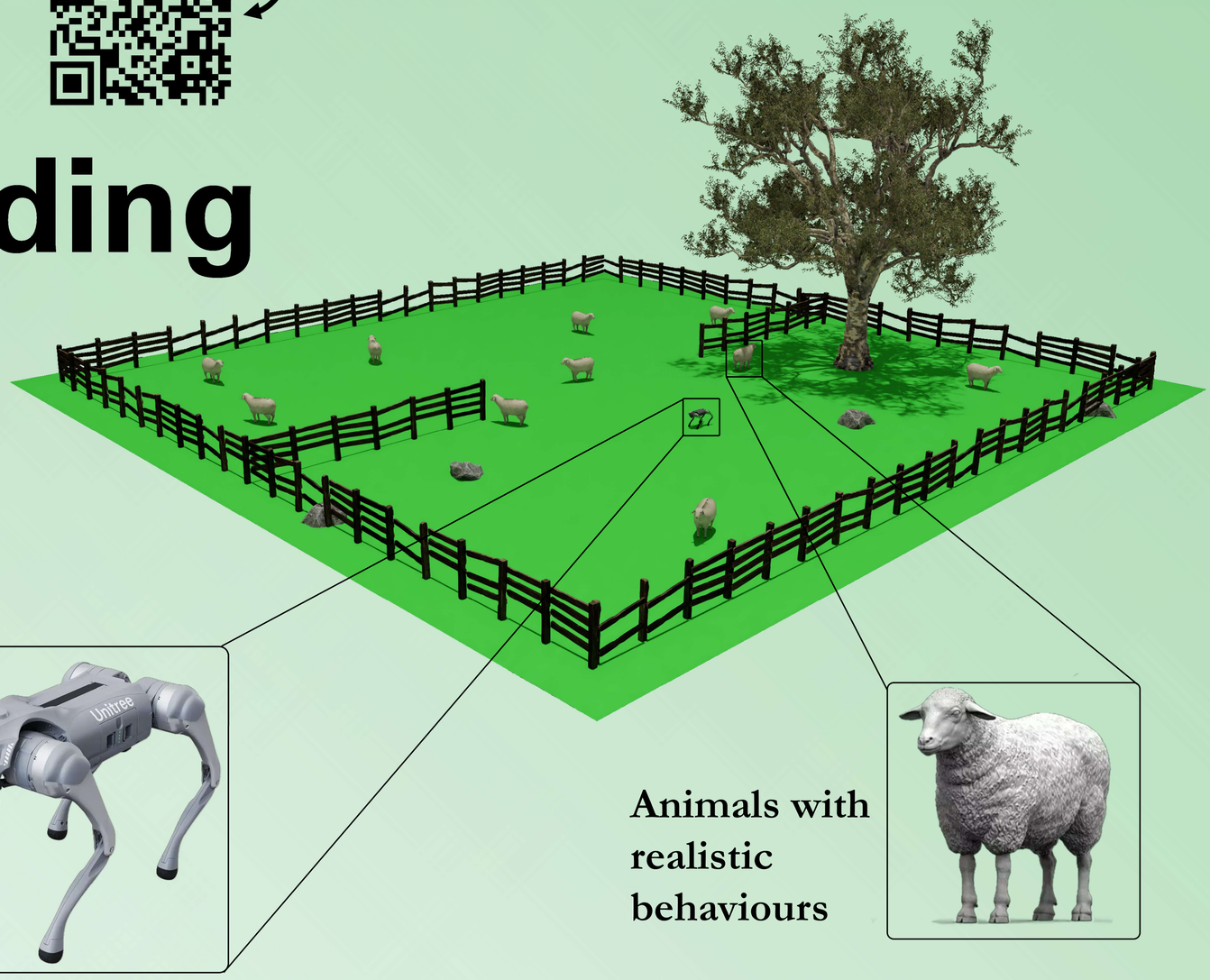
Reinforcement Learning

Scan me

# Livestock Herding Simulation Environment




## 1 Overview

The project aims to support the School of ICT's research on training robotic platforms for efficient livestock herding by developing a modular and highly-configurable livestock herding simulation environment. This provides a cost-effective strategy for training RL policies in a realistic virtual environment, prior to their fine-tuning and deployment in the real-world. The project was implemented in Nvidia Isaac Lab using Python.

## 2 Key Features

### Configurability

- Highly configurable parameters
  - Including paddock layout and per-species behaviour
- Configurable terrain
  - Procedurally generated with configurable elevation points
- Modular architecture enabling extensibility
  - Easy species swapping

### Realistic Behaviour

- Utilises Nvidia Isaac Sim physics engine
- Simulated flocking using Boids rules
- Obstacle avoidance
  - Includes experimental algorithms for intelligent obstacle navigation
- Realistic interactions between herding robot and animals

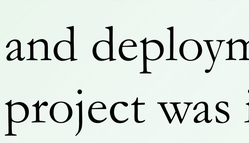
### RL API

- Supports a standard reinforcement learning API for policy training
- Features a simplified 'standard' simulation environment for rapid experimentation, and a 'vectorized' version for efficient GPU-native policy training

## 3 System Design

### Environments

Standard environment for quick experiments.



Vectorized environment for RL policy training.

### Isaac Lab

Physics and rendering engine.

### Robot

Herding robot can be trained to herd the automatons.

### Automatons

Animals (e.g., sheep) with realistic behaviours.

### Config Parameters

Config parameters control the behaviour of the simulation and facilitate extensive flexibility.

### Obstacles

Fences and other obstacles constrain movement within the environments.

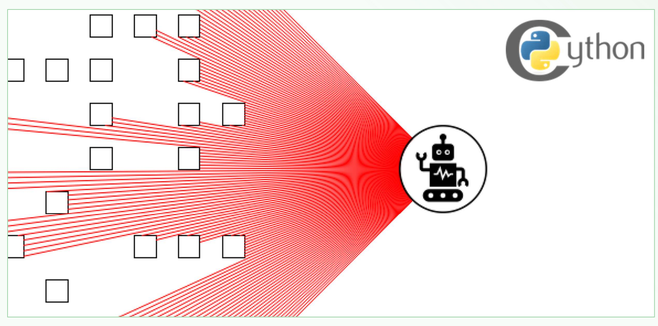
## 4 Derivative Products

### Efficient Multithreaded 2D Analytical Raycasting Library for Python

Developed to support intelligent obstacle navigation algorithms, though not used in the final product. Environment obstacles are stored as collections of linear line segments, allowing intersections with rays to be efficiently computed explicitly. Implemented in Cython to leverage CPU multithreading.

### Benchmarks

Mean runtime: 25 ms
- 800 obstacles
- 128 rays




Shreyansh Adhikari  Tim Cox  William Ditchfield  Dipesh Giri  James Stansfield