PlayMeow: Machine Learning for Pet-Interactive Robotic Systems

*Teaching Robots How to Love

Miles Popiela
Computer Science
Virginia Commonwealth University
Richmond, USA
popielamc@vcu.edu

***Abstract*—This paper presents an approach to robot-pet interaction through the implementation of a reinforcement learning-based control system for a laser-guided robotic turret. The system utilizes Convolutional Neural Networks (CNN) to process real-time 3D/2D positional data of Senna and learn optimal movement patterns for engagement and stimuli. By combining mixed-reality path programming for training and physical hardware for deployment, I demonstrate an adaptive system capable of natural interaction with pets while maintaining safety parameters. I hope to show improvements in engagement metrics and movement efficiency compared to boring programmed behaviors.**

***Index Terms*—Mixed-Reality, Robotics, Reinforcement Learning, CNN, Human-Animal-Robot Interaction, Pet Technology**

I. Introduction

The field of human-robot interaction has expanded significantly in recent years, with applications spanning healthcare, education, and domestic environments. However, comparatively limited attention has been dedicated to robot-animal interaction, particularly in household settings where pets constitute important family members. This research gap presents a unique opportunity to explore how intelligent systems can enhance the quality of life for domestic animals through adaptive engagement.

This paper presents Playmeow, an intelligent robotic system designed to interact with feline subjects through laser-based play. Using my cat Senna as the primary test subject, this research develops a reinforcement learning framework that adapts to individual feline preferences and behaviors. The system leverages real-time 3D positional data captured through SteamVR tracking technology to create personalized and engaging play experiences.

Playmeow processes time-series movement data through a sophisticated neural network architecture that includes:

- A sliding-window approach for capturing temporal movement patterns
- Multi-feature input vectors incorporating relative positioning, velocity, and engagement indicators
- A four-layer neural network producing normalized movement vectors
- A carefully structured reward system that reinforces positive engagement

The research contributes to both animal-computer interaction and machine learning domains through three key innovations:

First, a comprehensive data processing pipeline transforms raw positional data into predictive features that capture feline behavioral patterns. Second, a hybrid learning approach combines behavioral cloning with reinforcement learning, allowing the system to first learn from human demonstrations and then refine its behavior through direct interaction. Third, a training methodology integrates Unity-based simulation environments with physical hardware deployment, creating a safe and effective development pathway.

Unlike previous pet-interactive systems that rely on predetermined movement patterns, Playmeow implements a progressive learning protocol that evolves from human demonstration to autonomous operation. The system addresses practical implementation challenges including tracker attachment, safety mechanisms, and engagement assessment while maintaining rigorous evaluation through both quantitative metrics and qualitative behavioral assessment.

This research represents a significant step toward creating adaptive robotic systems capable of meaningful interaction with domestic animals, potentially extending beyond entertainment to therapeutic and enrichment applications.

II. Problem Statement

The development of responsive robotic systems for pet interaction presents several key challenges:

- Real-time processing and response to dynamic pet behavior
- Translation of 3D/2D positional data into meaningful movement commands
- Development of appropriate reward functions that encourage engaging but safe interaction
- Integration of physical hardware constraints with learned behaviors
- Balance between exploration of new movement patterns and exploitation of known effective behaviors

The primary focus of this research is the development of a reinforcement learning framework that can effectively process continuous 3D positional input and generate appropriate movement commands for a robotic turret system, while adapting to individual pet behavior patterns.

III. Background and Related Work

A. Pet-Robot Interaction Systems

The intersection of robotics and animal interaction has emerged as a promising research area, with applications spanning entertainment, enrichment, and therapeutic interventions. Recent advancements in computer vision, machine learning, and sensing technologies have enabled more sophisticated systems capable of detecting and responding to animal behaviors in real-time. Several foundational approaches exist in the literature that inform the current work. Huang and Yang [1] developed PiCat, an automated laser toy system that integrated YOLOv3-tiny for cat detection with predetermined laser movement patterns. Their system activated when a cat was detected and directed the laser near the edge of the cat's bounding box to avoid eye contact. While achieving reasonable processing speeds (approximately 0.6 seconds per image), their implementation relied on fixed behavioral patterns without adaptive learning capabilities. Building upon detection-based systems, Sieck [2] advanced the field with a safety-oriented approach that incorporated computer vision to establish prohibited zones for laser operation. His YOLO-based implementation demonstrates the feasibility of incorporating safety constraints within pet-interactive systems, a critical consideration for any deployment in domestic environments. However, similar to Huang's work, Sieck's system lacked the capacity to adapt to individual pet preferences or evolve interaction patterns over time.

B. Machine Learning in Gaming

The application of machine learning in gaming contexts provides valuable insights for interactive robotic systems. Machine Learning Arena (MLA) represents a significant exploration of using imitation learning to create responsive game agents. In MLA, players train a virtual boxer by demonstrating techniques that the ML agent then learns to replicate [3]. The system employs Behavioral Cloning (BC), a form of supervised learning that copies human player behavior through observation of input-output pairs. This approach allows the MLA system to develop individualized fighting styles based on each player's unique training.

One particular challenge noted in the MLA project was effectively communicating machine learning progress to users without technical backgrounds. The developers created multiple visual indicators including a "Copy Meter" and "Train of Thought" visualization to represent the learning progress of their boxing agent. These UI solutions sought to bridge the gap between complex ML concepts and intuitive player feedback, a challenge directly relevant to pet-interactive systems where transparency in learning would enhance engagement.

Similar to MLA's approach, Playmeow employs machine learning to create adaptive interactions, though with Senna's rather than human players. While MLA utilized Behavioral Cloning to mimic human inputs, our system processes positional data through Convolutional Neural Networks to learn optimal engagement patterns. Both systems face the common challenge of balancing computational complexity with real-time responsiveness, though Playmeow must additionally account for the unpredictable nature of animal subjects.

C. Reinforcement Learning and Imitation Learning for Interactive Systems

The integration of reinforcement learning (RL) and imitation learning offers powerful advantages for developing adaptive robotic systems. As demonstrated by Karavaev et al. [4], the simultaneous application of these approaches can significantly enhance training efficiency and create more natural behaviors in AI agents. In their study, an AI tank agent initially trained with only reinforcement learning required 10 hours to effectively combat two opponents. However, after implementing imitation learning alongside RL, similar performance was achieved in just 40 minutes, with the ability to engage 3-4 opponents simultaneously.

Reinforcement learning operates on the premise that an agent exists within an environment and selects actions from a set of possibilities. The environment responds by changing state, and the agent receives a reward or punishment signal. Through this interaction, the agent learns to optimize its strategy for maximum reward [4]. This approach is particularly effective for balancing short-term and long-term benefits, making it ideal for systems that must adapt to dynamic environments such as pet interaction scenarios.

Complementing RL, imitation learning employs a teacher-student paradigm where the student agent observes and attempts to replicate the actions of a teacher (which could be a human, another neural network, or a deterministic algorithm). This approach is especially valuable when developing systems that should exhibit natural behavior patterns rather than machine-like perfection [4]. In the context of pet-interactive systems, imitation learning can help create movement patterns that appear more organic and engaging to animals.

D. Engagement Metrics and Adaptive Systems

A key challenge in pet-interactive systems is defining and measuring engagement. Mancini et al. [5] established foundational principles for Animal-Computer Interaction (ACI) that emphasize the importance of species-appropriate design and behavioral indicators of engagement. Their work highlights the ethical considerations and methodological approaches necessary for developing technologies that genuinely benefit animal subjects. These principles provide valuable frameworks for defining appropriate reward functions in reinforcement learning systems designed for animal interaction. The current research synthesizes these approaches, extending beyond detection-based systems to incorporate adaptive learning capabilities that respond to individual pet behaviors. By combining the safety considerations pioneered by Sieck with the detection capabilities demonstrated by Huang and Yang, and extending these with reinforcement learning techniques inspired by work in human-robot interaction, Playmeow represents a significant advancement in pet-interactive technology. Unlike previous implementations that rely on predetermined movement patterns, this system leverages multiple prediction horizons to anticipate Senna's behavior while continuously adapting based on engagement metrics. This approach not only enhances the interactive experience but also creates opportunities for personalized enrichment tailored to individual pet preferences.

IV. Methodology

A. System Architecture

The Playmeow system integrates hardware and software components into a comprehensive pipeline:

- **Data Acquisition Hardware:**
  - Two SteamVR Basestations providing spatial tracking coverage
  - Two HTC Vive trackers (one attached to Senna, one to the turret) for real-time 3D positioning
  - One Raspberry Pi serving as the central processing unit
  - Two servo motors enabling 2-DoF movement for the laser turret
  - Custom 3D-printed assembly housing the laser emitter and camera
- **Software Components:**
  - Unity-based simulation environment for preliminary training and evaluation
  - Custom data processing pipeline for feature extraction
  - Neural network models implemented using TensorFlow/PyTorch
  - Safety constraint management system

The system begins with accurate positional tracking of both Senna and the laser dot. This tracking data feeds into the machine learning pipeline, which processes the information and generates appropriate movement commands for the robotic turret system.

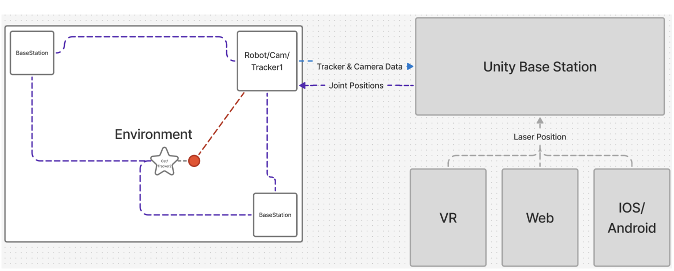

Fig. 1. Hardware setup showing SteamVR basestations, Vive trackers, Raspberry Pi, servo motors, and 3D printed assembly.

B. Data Processing Pipeline

Raw positional data undergoes systematic transformation to extract meaningful behavioral patterns:

- Time-series segmentation using sliding windows (10 timesteps) creates sequential input samples
- Calculation of derived features including velocity and acceleration vectors
- Feature normalization using StandardScaler ensures consistent model training
- Implementation of 5-timestep prediction horizons enables anticipatory positioning

This pipeline transforms simple positional coordinates into rich feature sets capturing movement patterns and behavioral trends, significantly enhancing the system's predictive capabilities.

C. Neural Network Architecture

Based on the requirements of the pet interaction system and informed by approaches in similar domains [4], I selected the following architecture (subject to change):

- A feed-forward neural network with 3 hidden layers (128, 256, 128 units respectively)
- ReLU activation functions for hidden layers and tanh for the output layer
- Output layer: 2 neurons representing normalized movement vectors in X and Z directions (range [-1,1])
- Key input features:
  - Senna's position (X,Z) relative to the current laser position
  - Senna's velocity vector (speed and direction)
  - Binary engagement indicator (based on head orientation and movement patterns)
  - Time since last successful engagement
  - Distance from laser to environmental boundaries
- Training parameters:
  - Batch size: 32
  - Learning rate: 1.0×10^{-3} with decay
  - Discount factor (gamma): 0.99

The network architecture prioritizes real-time responsiveness while maintaining sufficient complexity to learn engagement patterns. By outputting normalized movement vectors rather than absolute coordinates, the system can produce context-aware responses that adapt to Senna's position and behavior. All input features are normalized to the [-1, 1] range to ensure balanced feature importance and training stability.

Initial experiments with both larger and smaller network configurations indicated that this architecture provides an appropriate balance between computational efficiency and learning capacity for the specific requirements of pet interaction.

D. Reward System

The reinforcement learning framework implements a carefully designed reward mechanism:

- High engagement detection: +0.1 (when Senna actively pursues the laser)
- Low engagement periods: -0.2 (when Senna loses interest)
- Successful play session completion: +0.4 (when engagement is maintained for the target duration)

- Premature session abandonment: -0.3 (when Senna leaves the play area before session completion)
- Maintaining appropriate distance from Senna: +0.01 per second
- Operating in prohibited zones: -0.1 per occurrence

This reward structure encourages the development of movement patterns that maintain pet interest while respecting safety boundaries.

E. Hybrid Learning Approach

The Playmeow system implements a three-phase hybrid learning methodology:

1) Phase 1: Behavioral Cloning: Human operators demonstrate effective laser movement patterns while the system records Senna's positional data, laser trajectories, and engagement metrics. The neural network learns to replicate successful human-guided interaction patterns through supervised learning of these input-output pairs.

2) Phase 2: Simulation Refinement: Initial models are deployed in a Unity simulation environment where virtual laser patterns interact with Senna's recorded movement data. This allows safe optimization of movement parameters without risking pet disengagement or fatigue.

3) Phase 3: Reinforcement Learning: The pre-trained model is deployed to physical hardware and continues learning through direct interaction. Real-time reward signals derived from Senna's engagement metrics guide further optimization of movement patterns.

The transition between phases follows a progressive responsibility protocol:

1) Initial complete human control with system observation
2) Gradual system assumption of control over high-confidence movement patterns
3) Autonomous operation with periodic human intervention when engagement declines

F. Training Process

1) Data Collection Protocol:

- 5-minute calibration period establishing baseline movement patterns and environmental awareness
- 15-minute active play session with human-controlled laser patterns
- Mandatory 10-minute rest periods between sessions to prevent pet fatigue
- Data collection at 50Hz yielding approximately 45,000 raw state samples per 15-minute session
- Downsampling to 5Hz for training efficiency, resulting in 4,500 state-action pairs per session [maybe]

2) Data Processing: The raw positional data undergoes targeted processing to enhance training efficiency:

- Sliding window approach with 80% overlap between consecutive training samples creates temporal coherence while maximizing usable data following methodology similar to Wen et al. [6] who demonstrated its effectiveness in time series applications
- Simple noise filtering to remove tracking irregularities and jitter
- Input normalization to standardize all features within the [-1, 1] range, ensuring balanced feature importance during training
- Validation protocol that randomly withholds 20% of collected sessions for testing to prevent overfitting

The initial dataset comprised approximately 45,000 raw state samples collected across multiple play sessions with Senna. After preprocessing and quality filtering to remove periods of inactivity or technical disruptions, the final training dataset contained approximately 36,000 valid state-action pairs.

3) Training Implementation: Training proceeded with the following parameters:

- 5-fold cross-validation to ensure model generalization
- Early stopping with patience=10 to prevent overfitting
- Learning rate decay (factor=0.1) applied when validation loss plateaued
- Training performed on NVIDIA RTX 3090 GPU with 10GB VRAM

G. Implementation Considerations

1) Tracker Attachment:

- Lightweight harness modified to securely attach Vive tracker
- 3-5 day habituation protocol for Senna to get accustom with the harness
- Maximum 20-minute session duration preventing discomfort

2) Safety Mechanisms:

- Software boundaries preventing laser operation near Senna's eyes
- Real-time monitoring system ensuring appropriate distance maintenance
- Emergency stop functionality triggered by anomalous behavior detection

H. Evaluation Framework

1) Quantitative Metrics:

- **Engagement Duration:** Average time Senna actively tracked/pursued the laser pattern
- **Response Latency:** Time between changes in Senna's behavior and corresponding system adaptation
- **Path Efficiency:** Ratio of successful engagement triggers to total movement distance
- **Learning Curve:** Improvement in engagement metrics over successive sessions

2) Qualitative Assessment: A panel of three veterinary behaviorists evaluated video recordings of play sessions, rating:

- Naturalness of interaction patterns
- Signs of positive engagement versus frustration
- Appropriateness of activity level for different temperaments of Senna
- Comparison to human-controlled play sessions

V. Future Work

Planned developments include:

- Implementation of 3D environmental parameters
- Integration of pet behavior prediction rather than just sensing and visualization
- Extension to multiple pet scenarios
- Long-term interaction pattern analysis
- PointCloud multi-camera system

VI. Results

[To be completed after experimental results]

VII. Conclusion

[To be completed after experimental results]

References

[1] K. Huang and Y. Xu. (2021, May) Automatic cat laser toy. Cornell University ECE 5725 Spring 2021 Projects. [Online]. Available: https://github.com/FrankHuangHKP/CatLaser

[2] D. Sieck, "Enhancing feline exercise: A safe yolo-based laser toy," Master's thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, July 2023, master's thesis in Automatic Control and Robotics.

[3] G. Ferguson, J. Cattelona, J. Kreiselman, and K. Corry, "Machine learning arena: Creating an ml based game," *Worcester Polytechnic Institute*, 2019.

[4] V. Karavaev, T. Kiseleva, and O. Orlinskaya, "Simultaneous use of imitation learning and reinforcement learning in artificial intelligence development for video games," in *Proceedings of REMS 2018–Russian Federation & Europe Multidisciplinary Symposium on Computer Science and ICT*, 2018.

[5] C. Mancini, "Animal-computer interaction: a manifesto," *Interactions*, vol. 18, no. 4, p. 69–73, Jul. 2011. [Online]. Available: https://doi.org/10.1145/1978822.1978836

[6] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," *arXiv preprint arXiv:2002.12478*, 2020.