Teaching AI the Cat Righting Reflex:

A Deep Reinforcement Learning Approach

Raine Cui

December 2, 2025

1 Introduction

The ability of cats to rotate mid-air and land on their feet has fascinated scientists for over a century. This phenomenon is particularly interesting because cats achieve rotation while conserving angular momentum, starting and ending with zero rotational momentum.

From a computer animation perspective, this represents a challenging problem of creating believable character animation for scenarios that are difficult or unethical to capture through motion capture, and such data could not adapt to novel situations in real-time.

In this project, we address these limitations through a physics-based simulation approach powered by deep reinforcement learning (DRL). Our approach integrates three key components: 1) a rigid body dynamics model based on the flexible spine hypothesis, 2) skeletal rigging to bridge physics simulation and visual rendering, and 3) a reinforcement learning framework that discovers physically plausible control policies through interaction with the simulated environment.

Our main contributions are:

1) A simplified yet effective physics-based model using two rigid bodies and a 3-DOF spherical joint.
2) A skeletal rigging system that separates efficient physics computation from visual rendering.
3) Insights into reward shaping for this DRL task, including identification and resolution of local optima through iterative design.

2 Related Work

Physics-Based Character Animation. Physics-based approaches to character animation have a long history in computer graphics. Methods like SIMBICON[1] demonstrated that simple control policies could produce stable biped locomotion through rigid body simulation. These approaches ensure physical plausibility that all generated motions automatically respect physics laws such as momentum conservation and collision dynamics. Our work follows this tradition, using rigid body dynamics and joint constraints to model the cat's flexible spine.

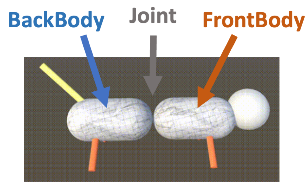
Deep Reinforcement Learning for Animation. Recent advances in deep reinforcement learning have enabled learning of complex motor skills for simulated characters. DeepMimic[2] showed that RL agents could learn acrobatic movements by imitating reference motions while maintaining physical plausibility. Unlike motion capture replay, DRL-based approaches can generalize to new situations. Our work applies similar DRL techniques but focuses on a task where reference motion is impractical to obtain.

Cat Biomechanics. The cat righting reflex has been studied extensively in biomechanics literature. Kane and Scher[3] provided a dynamical explanation showing that cats primarily use their flexible spine to redistribute angular momentum between body segments. While the total angular momentum remains zero, strategic bending and twisting of the spine allows net rotation. More recent studies confirm that the spine's flexibility, rather than leg movements or tail use, is the primary mechanism. Our physical model is based on this understanding, focusing on spinal degrees of freedom.

3 Method

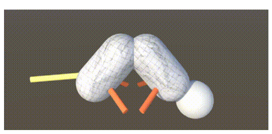
3.1 Physical Model Design

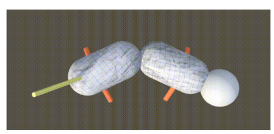
Our model simplifies the cat's anatomy while preserving the key biomechanical principle: a flexible spine connecting two body segments. We represent the cat as two rigid body capsules connected by a spherical joint:

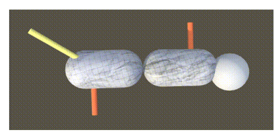


The joint has three degrees of freedom (DOF) control:

1) **Spine Bending** allows the cat to arch/flatten its back (range: 0° to 100°)
2) **Lateral Bending** allows side-to-side bending motion (range: ±50°)
3) **Axial Twist** allows rotation around the spine axis (range: ±120°)


DoF 1: Spine Bending (0° – 100°)


DoF 2: Lateral Bending (±50°)

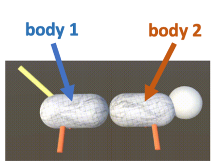

DoF 3: Twisting (±120°)

The joint is actuated through target angular velocity control instead of direct torque application, using Unity's ConfigurableJoint motor system. To maintain numerical stability in the physics engine, we apply moderate air drag. We verified that the learned agent can still successfully flip with drag disabled, showing that the strategy is based on internal momentum redistribution rather than "swimming" through air resistance.

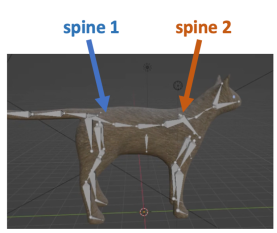
The legs and tail are included as decorative visual elements but do not participate in physics calculations. This simplification is justified by biomechanics research indicating that the flexible spine is the primary mechanism for the righting reflex.

3.2 From Physics to Visual

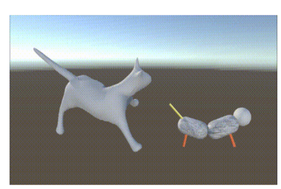
To achieve high-quality visual output while maintaining efficient physics simulation, we employ skeletal rigging to separate these concerns.



Physical Model



Rigged Model


Mapping between two Models

We obtained a rigged cat model with skeletal structure and skinned mesh. Using Unity's Animation Rigging system, we bind the rotation of the FrontBody capsule to a front spine bone, and the BackBody capsule to a rear spine bone.

When the physics simulation updates the capsule orientations, the Animation Rigging system propagates these rotations to the appropriate bones, and the skinned mesh deforms accordingly.

With this architecture, physics computation runs on simple primitives for speed and stability, while visual rendering uses a detailed mesh for aesthetic quality. The separation allows easy model replacement or refinement without changing physics

3.3 Reinforcement Learning Framework

We formulate the cat righting task as a Markov Decision Process and solve it using the Proximal Policy Optimization[4] (PPO) algorithm.

Observations. The observations are designed to mimic a real cat's awareness of its body. For each of the two body segments, we use local gravity direction (the direction of gravity in the body's local coordinate frame) and local angular velocity (the rotational velocity of that segment in its own frame). These observations are invariant to the cat's absolute position and orientation in world space and provides sufficient information to determine both the current misalignment (via gravity direction) and the rate of change (via angular velocity).

Policy Network. We use a multi-layer perceptron with 2 hidden layers of 32 units each and tanh activation functions.

Action Space. A 3-dimensional continuous action representing target angular velocities for the three joint DOFs.

3.4 Reward Design

The reward the sum of the following three components:

Alignment Reward. We measure how well each body segment is aligned with the belly-down orientation. For each segment, we compute the dot product between its "belly down" vector and the gravity direction vector, then transform this to a score using an exponential function: $score = \exp(\alpha * (alignment - 1))$. Here, alignment is normalized to [0, 1] before the exponential. The scale parameter $\alpha = 2$ sharpens the reward gradient near the goal. Since there are two body segments, we take the minimum of the two segment scores: $R = \min(score\_front, score\_back) - prevR$ to ensure that both segments must be aligned.

Efficiency Penalty. To encourage energy-efficient motion, we penalize the magnitude of the action vector, scaled by the current alignment score: $R = -0.05 * min\_score * ||action|| * \Delta t$. The scaling by min_score means the penalty increases as the cat gets closer to the goal, encouraging precise control near convergence while allowing aggressive motion when far from the target.

Angular Velocity Penalty. To prevent shaking and encourage smooth motion, we penalize the relative angular velocity between the two body segments: $R = -0.005 * min\_score * ||\omega\_front - \omega\_back|| * \Delta t$.

4 Implementation and Results

4.1 Training Details

We implemented the environment in Unity 6.0 using ML-Agents toolkit[5]. Training was conducted with 16 parallel environment instances, each starting with a random initial orientation. Episodes last 3 seconds (150 fixed timesteps at 50 Hz simulation rate), after which the environment resets regardless of success. Training converged after approximately 2 hours.

4.2 Results

The learned policy exhibits smooth, coordinated control of all three spinal DOFs to achieve rotation, and the motion appears natural without abrupt jerky movements.

A key advantage of this RL-based approach over motion capture is real-time adaptability. We tested the agent's response to mid-flight disturbances by manually rotating the cat during execution. The agent continuously adapts, treating each disturbed state as a new starting condition and adjusting its strategy accordingly. This demonstrates the policy's generalization capability that it has learned a robust mapping from body state to control, not a fixed trajectory.

4.3 Evaluation and Future Work

This project demonstrates the value of physics-based simulation combined with reinforcement learning for creating responsive character animation. By modeling the underlying physics and allowing an agent to discover control strategies through trial and error, we obtain animations that are both physically plausible and adaptive to novel situations. Our approach demonstrates that reinforcement learning can discover effective control policies for the cat righting reflex from scratch, without motion capture data or hand-designed controllers.

However, several limitations should be noted:

Model Simplification. Our two-body model is a significant simplification of real cat anatomy. While adequate for demonstrating the core mechanism, it lacks the multi-segment spine, articulated limbs, and tail that contribute to real cat agility. Adding more body segments would increase realism but also expand the action space, likely requiring longer training and more sophisticated exploration strategies.

Landing Dynamics. Our simulation ends once the cat achieves belly-down orientation. A complete system would include ground contact, impact absorption, and landing stabilization. These are interesting directions for future work.

Acknowledgement

I thank Professor Yin for valuable feedback on this project and my presentation. I acknowledge the use of the Unity ML-Agents toolkit and the rigged cat model from https://www.cgtrader.com/items/4590731/download-page .

References

[1] Yin, K., Loken, K., & van de Panne, M. (2007). SIMBICON: Simple biped locomotion control. *ACM Transactions on Graphics (TOG)*, *26*(3), 105.

[2] Peng, X. B., Abbeel, P., Levine, S., & van de Panne, M. (2018). DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, *37*(4), 1-14.

[3] Kane, T. R., & Scher, M. P. (1969). A dynamical explanation of the falling cat phenomenon. *International Journal of Solids and Structures*, *5*(7), 663-694.

[4] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

[5] Juliani, A., Berges, V.-P., Vckay, E., Gao, Y., Henry, H., Mattar, M., & Lange, D. (2018). Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*.