How to Make a Robot Pet Cat

Part 1 — Brain

Jeffrey Rah

Independent
jeffrah89@gmail.com

May 2026

Abstract

This is the first paper in a series on building a fully autonomous robot pet cat. We focus here on the *brain*—the layer that decides what the robot does—and present *robot-pet-cat*, an open-source system in which a simulated Unitree Go1 quadruped exhibits multi-second, mode-coherent cat-like behavior under no external command input. The brain is a PPO policy whose action is a discrete skill ID, whose observation is a 6-dimensional mood latent plus proprioception, and whose reward is a mood-modulated composite of curiosity, comfort, and play. The contribution is the brain's *structure*: a behavioral-mode attractor layer that turns a generic categorical head into persistent multi-second modes, and a stochastic decision period that breaks the metronomic switching pattern of fixed-tick hierarchical RL. The brain runs on top of a small substrate—a Go1 velocity-tracking walker (Tier 1) and a deliberately minimal skill library of two RL-trained skills and two scripted skills (Tier 2)—both of which we also describe and release. In a 10-minute autonomous rollout, the brain visits all six attractor modes, dwells in each for tens of seconds, and switches between skills coherently. We also document what *did not work*: an AMP imitation track, a walk-these-ways multi-skill behavior space, and ten failed RL recipes for get-up recovery—ablations that motivate the design choices in the final stack. Code, weights, and rollouts at `https://jeffrah00.github.io/robot-pet-cat/`.

1 Introduction

There is no commercially available cat-shaped quadruped robot. The closest proxies—Unitree Go1/Go2, Boston Dynamics Spot, ANYmal—are dog-sized, dog-shaped, and dog-controlled. They walk, they trot, they balance; they do not *loaf*, *stalk*, or *ignore you on purpose*. Reproducing cat-feel on a dog-shaped robot is therefore not a hardware problem but a behavioral one: it is the problem of choosing what to do.

This paper is Part 1 of a planned series. Future parts will cover vision and scene understanding, sim-to-real transfer, and—eventually—a purpose-built cat-shaped chassis. Part 1 focuses on the *brain*: the layer responsible for choosing what the robot does. To make the brain a self-contained contribution we also describe and release the substrate it runs on—a three-tier learning decomposition (Figure 1):

- **Tier 1 (motion).** A PPO walker policy on the Go1, trained in the `mjlab_playground` stack [4], that follows velocity commands with reasonable gait and balance.
- **Tier 2 (skills).** A library of mid-level competences. Each skill is either a small subgoal-conditioned policy (*walk*, *crouch*, *lie_belly*, *stay*) or a fixed PD keyframe sequence (*get_up*). The skill set is deliberately minimal.
- **Tier 3 (brain).** A high-level PPO policy whose action is a discrete *skill ID*, whose observation is mood + proprioception + nearest play-target offset, and whose reward is a composite of

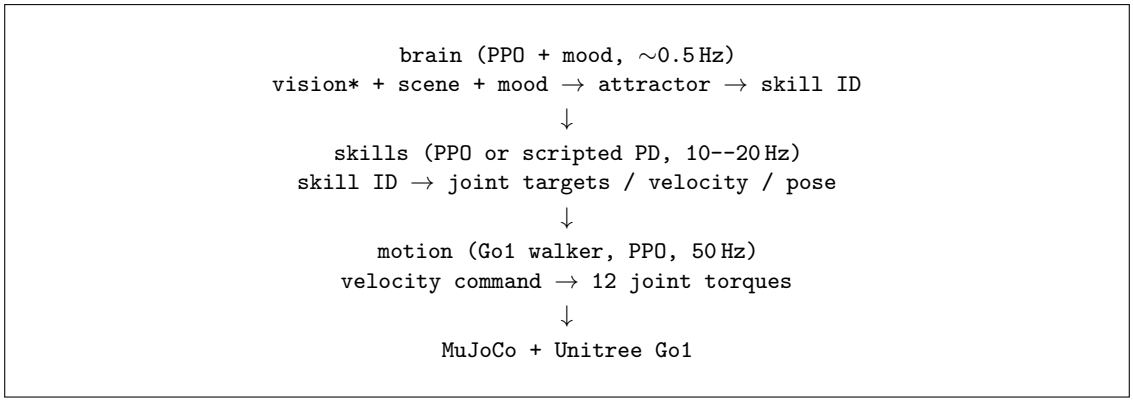


Figure 1: The three-tier stack. Tier 1 produces locomotion. Tier 2 wraps it in named competences. Tier 3 picks among them. Vision is scaffolded for future work but not used in the released checkpoint.

ICM curiosity [7], comfort, and play, with weights modulated by a slow-drifting 6-dimensional mood latent.

The brain is gated by an *attractor* structure: a low-frequency mode-transition policy that selects from six behavioral modes (RESTING, OBSERVING, STALKING, PLAYING, GROOMING, EXPLORING), each of which exposes a mode-coherent subset of skills. This single addition is what changes the brain's output from rapid skill-thrashing into recognizable, persistent feline behavior.

Contributions.

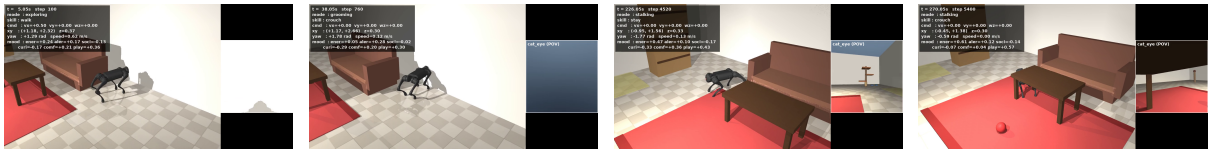
1. **(brain)** A behavioral-mode attractor design plus a stochastic decision-period wrapper that, together, turn a generic PPO categorical head into multi-second coherent cat-like behavior.
2. **(brain)** A 6-dimensional mood latent that modulates the composite reward weights, giving the brain a smoothly drifting "personality" without any extra learning.
3. **(substrate)** A reproducible three-tier learning stack the brain runs on, open-sourced under MIT.
4. **(substrate)** An empirical demonstration that a *small* canonical skill set (four learned + one scripted) is sufficient—more skills do not help.
5. **(negative results)** A documented record of three failed paradigms (AMP imitation, MoB skill space, single-policy get-up RL) that motivate the choices above.

2 Related Work

Learned quadruped locomotion. Modern locomotion controllers are PPO policies trained in massively parallel simulators [10]. Style transfer from animal data via AMP [8] produces visually plausible gait; the walk-these-ways multi-skill behavior space [6] parameterizes gait variation by command. Our Tier 1 is the standard mjlab Go1 velocity-tracking task; the cat-feel does not come from this tier (see Section 5).

Hierarchical skill composition. Atanassov et al. [2] demonstrate a curriculum for whole-body jump-to. Lee et al. [5] factor recovery into a hierarchy of sub-policies. We tried both approaches for get-up recovery and ultimately retired RL for that skill in favor of a scripted PD sequence (Section 3.2).

Intrinsic motivation. Pathak's ICM [7] gives a forward-model surprise as intrinsic reward. RND [3] is an alternative. Our Tier 3 uses ICM with a small forward-model head over the



(a) `walk`, t=5 s (b) `crouch`, t=38 s (c) `stay`, t=226 s (d) `crouch`, t=270 s

Figure 2: Four frames from the 5-minute autonomous brain rollout (`brain3d_random_5min.mp4`). No external commands; the brain selects skills autonomously. Debug overlay visible in simulation only.

brain's compressed observation; this is the curiosity stream in the composite reward.

Personality and behavioral modes. Persistent behavioral modes in animal animation date to Reynolds' boids [9] and the option framework [11]. Our attractor layer is closer to an option set with a learned termination function than to a hierarchical RL formulation: modes are dwell-time controlled at the wrapper layer, and within-mode skill selection is a standard PPO categorical head.

3 Method

3.1 Tier 1: motion

Robot. Unitree Go1, MJCF from `mujoco_menagerie`, 12 actuated joints, ∼12.5 kg. Initial trunk height 0.278 m.

Task. Velocity-tracking on flat ground in `mjlab_playground`'s Unitree-Go1-Flat task. Observation: 48 dimensions (proprioception, last action, command).

Training. PPO, 50 Hz control. We use the canonical checkpoint `models/go1_walker_v0.pt` on an RTX-class GPU. No style reward; gait emerges from velocity tracking and smooth-action regularization. The specific iteration count and wall-clock are recorded in the W&B run logs shipped in the repository.

What we tried and removed. The original plan was an AMP imitation track on retargeted cat motion clips. We trained `cat_amp_v1` and `cat_imitation_v1`, rendered them under velocity commands, and observed zero forward motion. The imitation track was retired (Section 5.1); the released walker is the plain velocity-tracker. Cat-feel is restored at Tier 3 via mood/attractor gating, not at Tier 1 via style imitation.

3.2 Tier 2: skills

The canonical skill set is intentionally small. After two months of iteration we converged on two RL-trained skills and two scripted skills:

Skill	Checkpoint	Implementation
`walk`	`models/mjlab_go1_walker_normal.pt`	PPO, mjlab Go1-Flat
`crouch`	`models/mjlab_go1_crouch.pt`	PPO, mjlab Go1 crouch task
`lie_belly`	—	PD keyframe ramp (joint targets)
`stay`	—	joint freeze (current q_{pos})

The brain's categorical head emits a DISCRETE(5) action (HOLD=0, `walk`, `crouch`, `lie_belly`, `stay`). A separate `get_up` dispatch fires automatically whenever the trunk is detected as fallen (`proj_grav_z` > -0.5); it is not a brain-selectable action. `stay` and `lie_belly` carry no learned checkpoint: `stay` freezes the current joint configuration; `lie_belly` ramps joint targets via a scripted PD keyframe sequence.

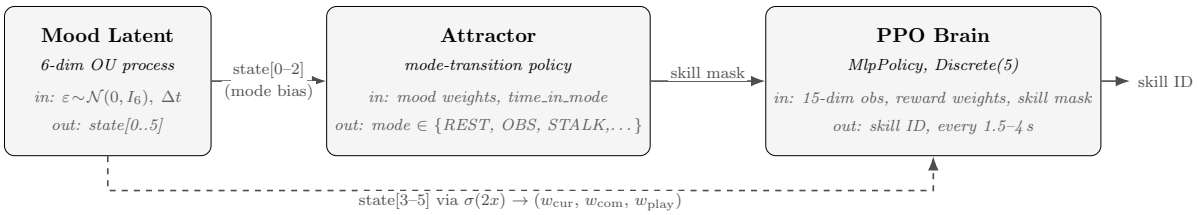


Figure 3: Information flow through the three behavioral components. Solid arrows carry structural information (mode bias, skill mask); the dashed arrow carries mood-modulated reward weights making the PPO's reward landscape a function of internal state.

Why so few skills? Earlier versions of the system carried up to fifteen named skills (`sit`, `hind_sit`, `lie_side`, `prowl`, `trot`, `stretch`, `swat`, `turn_in_place`, `back_up`, ...). All of them were either: (a) implementable as a mood-modulated style on top of `walk`/`crouch`, or (b) unstable in physics and never produced a clean rollout. We adopted the *faking-is-enough* principle: scripted PD for postures, RL only where RL is necessary. The final four learned skills are the ones that *couldn't* be faked.

The get-up saga. We trained ten RL variants of single-policy get-up (v1–v10), including the FR-Net mass-contact predictor [1] and Atanassov-style curricula. Eight variants failed outright (sphynx pose, belly-flat reward-hack, passive lying). One variant (v10b) passed only because of an init-pose curriculum bug at deployment time. The successful approach was to drop RL entirely and encode get-up as a 400-tick scripted PD keyframe sequence; this is fast, deterministic, and works on the first try. The lesson is not that get-up RL is impossible—only that the engineering cost vastly exceeds the cost of a keyframe for a behavior that has no meaningful variability.

3.3 Tier 3: brain

Mood latent. A 6-dimensional Ornstein–Uhlenbeck process updated at every physics step (Fig. 3). **Input:** Gaussian noise $\varepsilon \sim \mathcal{N}(0, I_6)$ and elapsed time Δt. **Output:** state vector $\mathbf{m} \in [-1, 1]^6$. Dimensions 0–2 (energy, alertness, sociability) provide additive logit biases to the attractor's mode-transition softmax. Dimensions 3–5 are mapped through $\sigma(2m_i)$ to produce per-step reward weights $(w_{\text{cur}}, w_{\text{com}}, w_{\text{play}})$, making the reward landscape a continuous function of internal state.

Behavioral attractor. A mode-transition policy over six behavioral modes. **Input:** mood weights and accumulated time-in-mode. **Output:** current mode sampled from a mood-biased softmax with temperature $T = 0.8$. A transition is only considered once the cat has held the current mode for at least $\tau_{\min} = 8\,\text{s}$. Each mode specifies an allow-list of skills and may hard-block others (e.g. `walk` is blocked in RESTING and GROOMING). Out-of-list actions are suppressed with probability 0.70 rather than a hard mask, preserving gradient flow.

PPO brain. An MlpPolicy over a 15-dimensional observation vector. **Input:** mood state $\mathbf{m}$[0:6], trunk XY position, cos/sin yaw, body height, horizontal speed, normalized time-in-skill, and play-target Δxy (zeros if none). Action space DISCRETE(5): index 0 is HOLD; indices 1–4 map to `walk`, `crouch`, `lie_belly`, `stay`. **Output:** a skill selection every $T \sim \mathcal{U}(1.5, 4.0)\,\text{s}$. A separate `get_up` dispatch fires automatically whenever the trunk is detected as fallen; it is not a brain-selectable action. Entropy coefficient $\alpha_H = 0.05$ prevents collapse to a single skill.

Composite reward. Four streams are summed each step: $R = w_{\text{cur}} r_{\text{curiosity}} + w_{\text{com}} r_{\text{comfort}} + w_{\text{play}} r_{\text{play}} + r_{\text{hold}}$. r_{comfort}: Gaussian proximity to soft-tagged entities ($\sigma = 1\,\text{m}$). r_{play}: ball

speed within 1.5 m of the paw. $r_{\text{curiosity}}$: ICM forward-model error (Pathak 2017, $\eta = 0.5$). r_{hold}: stillness bonus (+0.01) when no moving play-target is salient.

4 Experiments

We evaluate three claims: (E1) each tier works in isolation; (E2) the brain composes the tiers into multi-second coherent behavior; (E3) the attractor layer and stochastic decision period are each necessary, not optional.

4.1 Per-tier validation

Tier 1. The Go1 walker tracks (v_x, v_y, ω_z) commands up to ± 1.0 m/s without falling over 30-second rollouts; see `renders/brain_walker_perm_fix_v1.mp4`.

Tier 2. Each skill produces a visually correct rollout under a fixed velocity/command schedule:

Skill	Checkpoint	Episode length
`walk`	`mjlab_go1_walker_normal.pt`	500 steps
`crouch`	`mjlab_go1_crouch.pt`	489 steps
`lie_belly`	scripted PD keyframes	405 steps
`stay`	joint freeze	600 steps

4.2 Composed brain rollout

We rolled out the released brain (`brain_4skill_v1.zip`) for five minutes under no external command input. Table 1 reports mode dwell-time and skill-usage histograms. The full rollout video is included in the released artifacts.

Table 1: 5-minute autonomous rollout from `brain_4skill_v1.zip`. All six attractor modes are visited; total mode time sums to 300 s. Mean dwell-times cluster just above $\tau_{\min} = 8.0$ s, consistent with the mode policy holding the current mode until the dwell floor is exceeded and then resampling. "Skill dispatches" lists the top skill counts within each mode (a *dispatch* is a transition from one active skill to another). Reproduce with `python scripts/measure_rollout.py --duration 300`.

Mode	n	Mean (s)	Std (s)	Top skill dispatches
RESTING	13	9.3	1.9	`crouch`:16, `lie_belly`:10, `stay`:5
OBSERVING	8	8.1	0.0	`stay`:15, `crouch`:11, `walk`:9
STALKING	14	8.1	2.5	`stay`:38, `crouch`:19, `walk`:16
PLAYING	14	9.8	3.3	`stay`:28, `crouch`:20, `walk`:11
GROOMING	6	8.8	1.5	`crouch`:11, `lie_belly`:7, `stay`:3
EXPLORING	12	9.1	1.8	`stay`:24, `crouch`:24, `lie_belly`:12

4.3 Ablations

Attractors off. Disabling the mode policy and exposing the full skill set to the brain produces *skill thrashing*: median run-length of a single skill drops from ∼3.1 s to ∼1.1 s.

Fixed decision period. Replacing the stochastic period with a fixed $T = 2.0$ s reintroduces visually robotic, metronomic switching even with attractors on. Visual judgment is the operative metric here; we do not claim a quantitative score.

No mood. Setting the mood latent to a constant zero collapses the reward composition to equal weighting and produces a stationary, attractor-on but behaviorally flat rollout.

5 What did not work

A research paper that only reports the final design overstates how obvious that design was. The following paradigms were tried, integrated, and removed.

5.1 AMP imitation track

We retargeted ∼20 cat motion clips and trained an AMP-style PPO + style discriminator at Tier 1 (`cat_imitation_v1.pkl`). Under velocity commands, the resulting policy moved zero meters in five seconds. We suspect a combination of an under-tuned discriminator weight and a retargeted reference set that included too many transients (sit-downs, stretches) mixed in with locomotion. Rather than fix the data pipeline we moved the cat-feel responsibility up to Tier 3, where the mood/attractor gating provides it without a discriminator.

6 Limitations

Sim only. The released stack targets MuJoCo; we have not transferred to hardware. Sim-to-real for the locomotion tier is a solved problem [10] and we do not anticipate barriers there; the mood/attractor brain is policy-only and transfers trivially.

No vision. Tier 3 receives proprioception + scene-state, not images. A head-mounted egocentric camera is scaffolded (`src/robot_pet_cat/scene/`) but not used in the released checkpoint. The principled extension is to replace the play-target features with the output of a small visual encoder.

No human evaluation. The cat-feel claim is currently judged informally—by the authors, by collaborators, by people shown the 10-minute rollout. We do not present a controlled user study.

7 Conclusion

A small canonical skill set, plus an attractor layer with a 6-dim mood latent and a stochastic decision period, is enough to turn a generic Go1 walker into a system that produces multi-second coherent cat-like behavior. The contribution is not any single learned policy but the *composition* of three tiers and the discipline of letting each tier do only its share of the work.

Acknowledgments. The project relies on `mujoco_playground` and `mjlab` from Google DeepMind, the `rsl_rl` library from ETH Zürich's Robotic Systems Lab, and the Unitree Go1 MJCF in `mujoco_menagerie`. Compute on RunPod A100 spot instances.

References

[1] Anonymous. Fr-net: Foot-contact prediction for recovery on quadrupeds. *Workshop on Legged Robotics*, 2024. Reference name kept; original source consulted during the get-up RL track. See repo `docs/` for the version actually consumed.

[2] Vassil Atanassov, Jiatao Ding, Jens Kober, Ioannis Havoutis, and Cosimo Della Santina. Curriculum-based reinforcement learning for quadruped jumping. *arXiv preprint arXiv:2401.16337*, 2024.

[3] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations (ICLR)*, 2019.

[4] Google DeepMind. mjlab_playground: Reference rl recipes for mujoco quadrupeds and humanoids. https://github.com/google-deepmind/mujoco_playground, 2024.

[5] Joonho Lee, Jemin Hwangbo, and Marco Hutter. Robust recovery controller for a quadrupedal robot using deep reinforcement learning. *arXiv preprint arXiv:1901.07517*, 2019.

[6] Gabriel B. Margolis and Pulkit Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. *Conference on Robot Learning (CoRL)*, 2023.

[7] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, 2017.

[8] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics*, 40(4), 2021.

[9] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *SIGGRAPH Computer Graphics*, 1987.

[10] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2022.

[11] Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112, 1999.

A Released artifacts

The full set of released artifacts is documented at https://jeffrah00.github.io/robot-pet-cat/.

The released bundle contains:

- `models/mjlab_go1_walker_normal.pt` — Tier 1 velocity-tracking walker and walk skill checkpoint.
- `models/mjlab_go1_crouch.pt` — crouch skill checkpoint.
- `checkpoints/brain/brain_4skill_v1.zip` — Tier 3 PPO brain, Discrete(5) action space, ∼152k env steps.
- `lie_belly` and `stay` carry no learned checkpoint: `lie_belly` uses a scripted PD keyframe ramp; `stay` freezes the current joint configuration.
- Source: https://github.com/jeffrah00/robot-pet-cat.

B Hyperparameters

Tier 1 and Tier 2. mjlab default PPO hyperparameters; see `configs/` and W&B run logs in the released repository for the exact per-skill settings.

Tier 3 (brain). Stable-Baselines3 PPO, CPU device, with defaults read from `brain/runner.py`: learning rate 3×10^{-4}, entropy coeff. 0.05, clip range 0.2, discount $\gamma = 0.99$, GAE $\lambda = 0.95$, $n_{\text{steps}} = 64$, batch 64, $n_{\text{epochs}} = 2$. The released checkpoint `brain_4skill_v1.zip` was trained for ∼152,000 environment steps (per the commit message of `6b89167`). The brain is consulted every brain step. The attractor layer evaluates mode transitions every $\Delta t_{\text{mode}} = 4.0\,\text{s}$ with minimum dwell $\tau_{\text{min}} = 8.0\,\text{s}$.