

From LLM-Generated Specifications to Learned Quadruped Locomotion

Merve Atasever*, Keyan Azbijari*, Cagan Bakirci*, Alfredo Reina Corona*,
Tolga Izdas*, Richard Yang†, Erdem Biyik*, Jyotirmoy V. Deshmukh*

**Department of Computer Science
University of Southern California
Los Angeles, CA, USA*

{atasever, azbijari, cbakirci, reinacor, izdas, biyik, jdeshmuk}@usc.edu

*†University of Florida
Gainesville, FL, USA
yangrichard@ufl.edu*

Abstract—Quadruped robot locomotion policies are often trained using reinforcement learning, which in turn relies heavily on hand-crafted reward functions. Designing reward functions requires substantial manual engineering, and it is often unclear which local rewards will induce the desired global behavior. Shaped rewards from formal specifications in languages like Signal Temporal Logic (STL) can make rewards more interpretable, but writing STL specifications itself still requires domain expertise. We study whether large language models (LLMs) can fill this gap by generating Parametric Signal Temporal Logic (PSTL) specifications that are subsequently used for policy learning. Given a natural language locomotion objective and a constrained specification grammar, GPT-5.5 and Qwen 3.6 independently propose STL templates for command tracking, safety, and gait structure. We instantiate the parameters of the generated PSTL templates using expert trajectories and retain only specifications that are consistent with demonstrated expert behavior. The resulting specifications are then transformed into smooth, finite-history reward functions and used to train a quadruped locomotion policy with Proximal Policy Optimization (PPO) in MuJoCo XLA (MJX). We evaluate both *gait-aware* and *gait-agnostic* settings. The former specifies walking-trot, trot, and bound regimes, while the latter allows contact patterns to emerge from the task objective. We compare against hand-engineered rewards, Text2Reward-style LLM-generated reward code, and an expert-switching oracle. Gait-aware Qwen 3.6 specifications achieved 100% survival and command success across all tested speeds (0.3–2.1 m/s) and matched the target gait at high speeds, whereas Text2Reward achieved 0% for both metrics at ≥ 1.9 m/s. Videos: <https://stl-locomotion.github.io/>

Index Terms—quadrupedal locomotion, reinforcement learning, signal temporal logic, large language models, reward design

I. INTRODUCTION

Deep RL has enabled quadruped robots to learn agile locomotion from simulations and expert demonstrations, but performance still depends strongly on reward engineering [1]–[7]. A typical locomotion reward is a weighted sum of instantaneous tracking, posture, smoothness, contact, and energy terms. Although such rewards can produce performant policies, selecting various reward components and coefficients requires substantial domain knowledge and empirical tuning, with design choices often justified through ablations rather

than first principles [4], [8], [9]. Moreover, the resulting numerical, local rewards do not explicitly describe the global temporal behavior it is intended to induce. This is especially problematic for multi-gait locomotion: walking, trotting, and bounding differ not only in speed, but also in contact timing and support patterns. Alternatives such as inverse RL and preference-based RL reduce reliance on manually specified rewards by inferring objectives from demonstrations or human feedback, but introduce an additional reward-learning or optimization stage [10]–[15]. Our approach also uses expert trajectories, but only to calibrate and validate an explicit set of symbolic, global specifications whose structure is proposed by an LLM.

Another approach is to describe desired behavior explicitly using formal logic and convert specification-satisfaction into a learning signal. Prior work on logic-guided RL has leveraged Linear Temporal Logic (LTL) and Signal Temporal Logic (STL) for reward shaping and policy learning, while also exploring the inference of temporal-logic specifications from demonstrations, particularly in navigation and manipulation tasks. [16]–[21]. Formal specifications have similarly been incorporated into planning and control for legged robots, including locomotion over cluttered terrain, stability constraints, and STL-informed control for bipedal walking and push recovery [22]–[26]. For quadrupeds, however, comparable continuous temporal-logic formulations remain limited. Existing logic-driven gait-learning approaches largely employ discrete logical rules or reward machines defined over foot-contact propositions and operating primarily on contact-level logic and step-wise signals [27], [28].

Signal Temporal Logic or STL [29] is well suited to this problem because it expresses bounded time-horizon properties directly over real-valued trajectories and associates each specification with a quantitative robustness value measuring its margin of satisfaction or violation [30], [31]. These quantitative semantics makes it possible to use STL not merely as a *post hoc* verifier, but as a dense reward signal for RL [17], [19], [32], [33]. Unlike conventional rewards, an STL specification can explicitly encode temporal relationships such as how long

a condition should persist, how events should be ordered, or how contact patterns should evolve over a finite horizon.

In our previous work, we manually defined parametric STL templates for quadruped safety, command tracking, and gait behavior and calibrated their numerical thresholds from expert rollouts [34]. While this formulation improves interpretability and gait control relative to heuristic rewards, it leaves the burden of manual specification design unresolved.

Recent LLM-based reward-design methods address a closely related problem by generating executable dense reward functions directly from natural-language task descriptions [35], [36]. In parallel, language models have increasingly been used to translate natural language requirements into temporal logic representations [37]. We combine these ideas differently. Instead of asking an LLM to produce the numerical reward itself, we ask it to propose a symbolic temporal specification. Expert trajectories are then used to fit the specification parameters and reject candidates that disagree with demonstrated behavior. The retained specifications are subsequently converted into dense learning signals through STL robustness.

We investigate this design using Google’s Barkour quadruped. GPT-5.5 and Qwen 3.6 are prompted with the same locomotion objective and STL grammar, but are not provided numerical thresholds [38], [39]. Each model proposes Parametric STL (PSTL) specifications. We fit the values of these parameters from expert trajectories and retain only specifications that are consistent with those trajectories. During policy learning, we compute smooth trailing-window STL robustness online and use it as a PPO reward [40], [41]. We evaluate the approach in both *gait-aware* (multi-gait) locomotion, where speed regimes activate gait-specific specifications, and *gait-agnostic* locomotion, where no gait is prescribed and the robot is free to discover its own contact pattern.

Contributions.

- 1) We introduce a pipeline in which LLMs generate the *structure* of interpretable locomotion specifications while expert data grounds their numerical parameters.
- 2) We propose a simple expert-consistency filter: a generated specification is discarded when its median robustness on expert data is negative, preventing systematically violated LLM suggestions from entering the reward.
- 3) We evaluate both gait-aware and gait-agnostic formulations to study how explicitly prescribing gait structure affects the learned locomotion behavior.
- 4) We benchmark against a heuristic reward, Text2Reward, and an expert-switching oracle.

II. BACKGROUND AND PROBLEM SETTING

We consider Barkour vb in MJX [6], [7], [42]. At each control step ($\Delta t = 0.02$ s), the policy receives commanded linear and angular velocities ($v_x^{cmd}, v_y^{cmd}, \omega_z^{cmd}$), proprioceptive observations, the previous action, and a short observation history, and outputs 12 normalized joint-position commands. A PD controller maps these commands to actuator torques. The goal is to learn a policy $\pi_\theta(a_t | o_t)$ that tracks commands while remaining stable and energy efficient.

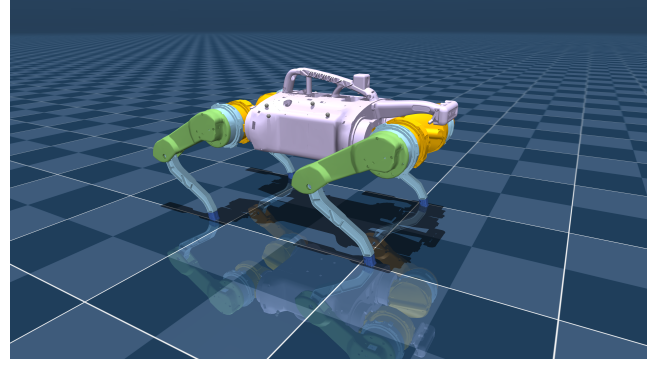


Fig. 1. Barkour vb robot in MJX.

An STL predicate has the form $f(x(t)) \geq 0$. Boolean operators combine predicates and temporal operators $\mathbf{G}_{[a,b]}$ and $\mathbf{F}_{[a,b]}$ respectively constrain predicate satisfaction to be globally true over the time interval $[a, b]$ or true at some point during the interval. Quantitative semantics assign a robustness $\rho^\varphi(x, t) \in \mathbb{R}$ to formula φ : positive values indicate satisfaction, and negative values indicate violation [30], [31]. Parametric STL (PSTL) replaces numerical constants in predicates or temporal intervals by parameters that can be fit from data [43].

We evaluate each temporal specification over the trailing window $\mathcal{W}_t = [t - H, t]$, and the resulting specification-level robustness values are subsequently combined using the smooth aggregation described in the next section [44].

III. LLM-GENERATED STL REWARD PIPELINE

A. Specification Generation

The LLM receives (i) a natural-language description of Barkour locomotion and (ii) an allowed STL grammar. We explicitly ask for *template structure only*: thresholds and temporal constants remain symbolic so that they can be estimated from expert data rather than invented by the language model. GPT-5.5 and Qwen 3.6 are queried independently with the same prompt.

For the multi-gait setting, the prompt describes three forward-speed regimes: walking-trot ($0 \leq v_x^{cmd} \leq 0.7$ m/s), trot ($0.7 < v_x^{cmd} \leq 1.7$ m/s), and bound ($v_x^{cmd} > 1.7$ m/s). Generated formulas can refer to shared tracking/safety signals and regime-specific contact statistics. Typical template families include examples shown below¹.

Let the trailing evaluation window at time t be defined as

$$\mathcal{W}_t = \{t - H, \dots, t\}, \quad (1)$$

where H denotes the robustness horizon. For an STL formula of the form $\mathbf{G}_{\mathcal{W}_t} \mu$, where μ is a predicate, the robustness at time t is computed as

$$\rho^{\mathbf{G}_{\mathcal{W}_t} \mu}(t) = \min_{k \in \mathcal{W}_t} \rho^\mu(k). \quad (2)$$

¹Complete sets of generated specifications and prompts are available on the project website: <https://stl-locomotion.github.io/>

Thus, the robustness is determined by the worst predicate margin observed within the trailing temporal window.

Example 1. Torque limits: For Barkour vb, the actuators provide a peak output torque of 18 Nm ($\tau^{\max}$) at each joint, hence motor torque limits must be bounded by this limit [6].

$$\varphi_{\tau} = \mathbf{G}_{\mathcal{W}} \left(\bigwedge_{j=1}^{12} |\tau_j| \leq \tau^{\max} \right). \quad (3)$$

The corresponding robustness is

$$\rho_{\tau}(t) = \min_{k \in \mathcal{W}_t} \min_{j \in \{1, \dots, 12\}} (\tau^{\max} - |\tau_j(k)|). \quad (4)$$

Example 2. Torso orientation: Roll and pitch are constrained using

$$\varphi_{\phi} = \mathbf{G}_{\mathcal{W}} (|\phi| \leq \phi_{\max}), \quad (5)$$

$$\varphi_{\theta} = \mathbf{G}_{\mathcal{W}} (|\theta| \leq \theta_{\max}). \quad (6)$$

Example 3. Center-of-mass (CoM) height: A minimum center-of-mass height is specified as

$$\varphi_z = \mathbf{G}_{\mathcal{W}} (z_{\text{com}} \geq z_{\min}). \quad (7)$$

Example 4. Support contacts: To require a minimum number of feet in contact with the ground, we use

$$\varphi_c = \mathbf{G}_{\mathcal{W}} (n_c \geq n_{\min}), \quad (8)$$

where n_c denotes the number of active foot contacts.

Example 5. Linear velocity tracking: For each commanded linear velocity component $i \in \{x, y\}$, tracking is specified as

$$\varphi_{v_i} = \mathbf{G}_{\mathcal{W}} (|v_i - v_i^{\text{cmd}}| \leq \epsilon_{v,i}). \quad (9)$$

The numerical parameters $\phi_{\max}$, $\theta_{\max}$, $z_{\min}$, $n_{\min}$, and $\epsilon_{v,i}$ are not generated by the LLM. The LLM determines only the symbolic structure of each STL specification, while the numerical parameters are subsequently estimated from expert trajectories. The resulting specification-level robustness values are then aggregated to construct the reward used for PPO training.

Multi-Gait vs Gait-Agnostic Policies Using the same gait for low speed and high speed motion may be inefficient or infeasible for quadrupeds [45], [46]. To determine transition velocity thresholds between speed regimes, we leverage the Froude number which is used to characterize gait transition speeds of different-sized quadrupeds: $\text{Fr} = \frac{v^2}{gh}$, where Fr is the Froude number, v is the linear velocity of the robot's torso, g is the acceleration due to gravity, and h is the maximum leg length [47]. Building on bio-inspired robotics gait design methodologies and reported gait structures that Barkour vb achieved to succeed, we leverage three speed regimes (modes): walking-trot, trot, and bound, selected based on commanded forward velocity $|v_x^{\text{cmd}}(t)|$. Using the Barkour robot's leg length of 0.41 m and the reference Froude numbers of 0.15 and 0.74 [48], we establish the respective gait transition velocity thresholds at 0.7 m/s and 1.7 m/s. Here, diagonal legs move

together for mid-speed regime (trot), whereas forelegs make contact approximately in phase, followed by the hind legs to achieve high speeds (bound).

While the multi-gait setting aims to learn a single policy that adapts its gait structure across speed regimes (walking-trot $\rightarrow$ trot $\rightarrow$ bound), the gait-agnostic setting does not enforce any target gait. We therefore use separate prompts for the two settings.

B. Robustness Reward

For active retained specifications, we aggregate multiple robustness terms using the soft-min $\text{smin}_{\beta}^{\text{sp}}(z_1, \dots, z_K) := \sum_{k=1}^K \pi_k z_k$, where $\pi_k = \frac{e^{-\beta z_k}}{\sum_{j=1}^K e^{-\beta z_j}}$. For the final reward shaping, each grouped robustness is normalized as $\tanh(\mu)$ to map to a value in $[-1, 1]$, preventing large robustness magnitudes from dominating the reward. The final scalar reward is:

$$r_t = w_{\text{safe}} \tanh(\mu_{\text{sa}}) + w_{\text{track}} \tanh(\mu_{\text{tr}}) + w_{\text{pattern}} \tanh(\mu_{\text{pa}}) \quad (10)$$

This decomposition makes the generated reward inspectable: during an evaluation we can report both the scalar RL reward and the robustness of each retained formula.

C. Data Grounding and Expert-Consistency Filtering

We collect 50 expert trajectories per regime, each containing 500 time steps, and compute the signals required by every generated template. Numerical parameters for PSTL templates are estimated from expert data statistics using quantiles appropriate to the predicate direction: upper quantiles for error ceilings and lower quantiles for safety floors.

Moreover, LLMs can propose technically sound but incompatible or overly strict requirements. We take advantage of expert trajectories to filter these specifications. Let

$$R_{\varphi} = \left\{ \rho^{\varphi}(x^{(i)}, t) \mid i = 1, \dots, 50, t \in \mathcal{T}_i \right\} \quad (11)$$

denote the set of robustness values obtained for specification φ across all expert trajectories and valid evaluation times. We retain

$$\varphi \text{ iff } Q_{0.50}(R_{\varphi}) \geq 0, \quad (12)$$

i.e., the specification is removed if its median robustness is negative. Equivalently, a formula that is violated on more than half of the expert evaluation points is not used for reward construction. We refer to this step as expert-consistency filtering.

IV. EXPERIMENTAL DESIGN

A. Training and Baselines

All learned policies use the same Barkour MJX environment, policy architecture, PPO implementation, command distribution, and domain-randomization scheme; only the reward source changes. We use friction randomization in $[0.6, 1.4]$ and actuator gain perturbations, following the existing training setup. During training, a new velocity command is randomly sampled every 10 s. The initial command range is set to

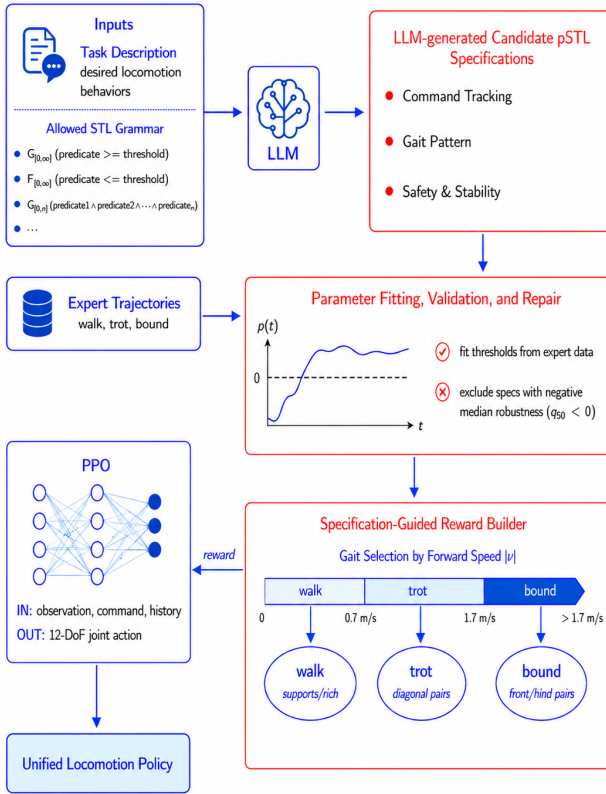


Fig. 2. Overall Pipeline.

$v_y^{cmd} \in [-0.2, 0.2] \text{ m/s}$, $\omega^{cmd} \in [-0.2, 0.2] \text{ rad/s}$, and the forward-velocity range is gradually expanded to $v_x^{cmd} \in [0.0, 1.9] \text{ m/s}$ as part of the curriculum for learning the locomotion at higher speeds. Details of the training procedure and domain randomization are provided in the GitHub repository.

We compare the following methods: (1) **Heuristic-reward** uses the standard hand-engineered locomotion reward in the original Barkour paper [6], (2) **human-STL** is the manually structured STL reward from the previous work and serves as an ablation for whether the LLM can replace manual specification design, (3 & 4) **GPT-STL** and **Qwen-STL** are each trained from the specifications generated by the corresponding LLM, (5) **Text2Reward** adapts direct LLM-generated reward code [35] (implemented using GPT-5.5 as the underlying language model), (6) **Expert-switching oracle** selects the corresponding specialized expert for the ground-truth commanded speed regime. It is not a learned unified controller and is reported as an oracle reference rather than a directly comparable training method.

B. Evaluation

We evaluate forward commands $v_x^{cmd} \in \{0.3, 0.5, 0.7, 1.0, 1.3, 1.6, 1.9, 2.0, 2.1\} \text{ m/s}$ with zero lateral/yaw command, using 20 rollouts of 500 steps per command and excluding the first 50 steps for tracking metrics. We leverage four evaluation

metrics: **cost of transportation (CoT)**, **survival rate**, **success rate**, and **gait match**.

We measure energy efficiency using cost of transportation $\frac{E}{m \cdot g \cdot d}$ where E is the total energy consumed, m is the robot mass, g is gravitational acceleration, and d is the planar distance traveled during the rollout. For episodes where the robot fails to survive, this metric is computed only up to the timestep of early termination. Lower CoT indicates better transport efficiency.

A rollout is counted as *survived* if the robot does not terminate early due to falling or violating the environment termination conditions.

We define success based on velocity tracking accuracy after the warm-up phase. Let v_x^{cmd} be the commanded forward velocity, and let $v_x^{loc}(t)$ denote the robot's forward velocity in its local body frame. The average post-warm-up forward speed for rollout k is $\bar{v}_x^{(k)} = \frac{1}{T-50} \sum_{t=51}^T v_x^{loc,k}(t)$. A rollout is considered successful if $|\bar{v}_x^{(k)} - v_x^{cmd}| \leq 0.15 |v_x^{cmd}|$, i.e., the average forward speed stays within $\pm 15\%$ of the commanded velocity.

Gait match is a qualitative label based on the dominant contact pattern: Success denotes a dominant pattern matching the prescribed gait, Partial denotes intermittent matching, and Fail denotes a different dominant pattern.

For each LLM, robustness horizon H is selected using validation performance from $\{1, 5, 10, 20, 30\}$.

V. RESULTS AND DISCUSSION

Table I shows that GPT-GA and Text2Reward-GA (GA denotes gait-agnostic) achieve the strongest quantitative performance in the gait-agnostic setting. They maintain command success and survival across the full evaluated velocity range while yielding the lowest cost of transport (CoT) for most commands. In contrast, the heuristic-reward policy loses tracking accuracy at the two highest speeds, and Qwen-GA loses both survival and success from 1.6 m/s onward. Despite the strong tracking performance of GPT-GA and Text2Reward-GA, visual inspection reveals a critical discrepancy between their quantitative results and their actual locomotion quality. The policies learn a bound-like contact pattern and exploits it across the entire tested speed range. At a low commanded velocity of $v_x^{cmd} = 0.3 \text{ m/s}$, the robot continues to move its legs at a high cadence, producing pronounced vertical body oscillations. While this behavior yields a 100% average velocity-tracking success rate, the resulting low-speed motion is visibly unnatural and undesirable. This highlights a fundamental limitation of evaluating locomotion solely through command success: high quantitative scores do not guarantee dynamically appropriate control. (Videos of the policies are provided on the project website).

This limitation underscores the value of the multi-gait formulation. When explicit gait structures are prescribed, the results in Table II exhibit a markedly different pattern. Human-STL is the most consistent overall gait-aware controller. Qwen-MG matches its command/survival performance and desired

bound behavior at high speed but fails to reproduce several prescribed lower-speed gaits. GPT-MG captures the low/mid-speed target gaits better but loses command tracking at 1.9 m/s and above. Text2Reward-MG is energy-efficient where it works but fails completely at high speeds. Ultimately, these results demonstrate that no single reward formulation universally dominates across both task settings and all evaluation criteria.

GPT-5.5 vs Qwen 3.6: The relative performance of the two LLM-generated specification sets depends strongly on whether gait structure is prescribed. In the gait-agnostic setting, GPT-GA is more robust at high speeds: it survives all tested commands and tracks commands accurately, while Qwen-GA terminates from 1.6, m/s onward. The comparison reverses in the multi-gait setting. Qwen-MG tracks commands throughout the full 0.3-2.1, m/s range and generally matches or improves upon GPT-MG in CoT, whereas GPT-MG fails the command-success criterion at 1.9, m/s and above. Thus, the quality of an LLM-generated specification cannot be characterized independently of the control setting in which it is used.

Why the Text2Reward comparison matters: Both Text2Reward and our approach begin from language-model knowledge, but the intermediate representation differs. Text2Reward directly commits to numerical executable reward code from current-step signals. Our pipeline first exposes the proposed behavior as symbolic temporal constraints, then fits thresholds from expert data. This comparison lets us ask whether exposing the LLM output as an STL specification changes the learned behavior relative to directly generating reward code.

The clearest difference between STL specification generation and direct reward-code generation appears in the high-speed regime. Text2Reward-MG terminates in every rollout at 1.9, 2.0, and 2.1 m/s, whereas Qwen-MG achieves 100% survival and command success at all three commands and exhibits the desired bound contact pattern. This advantage is regime-specific: Qwen-MG does not reproduce the prescribed gait at several lower-speed commands. The comparison indicates that the STL intermediate representation can produce a more effective high-speed learning signal than the evaluated Text2Reward baseline, while also showing that successful command tracking does not guarantee recovery of all intended gait structures.

Gait-aware vs gait-agnostic locomotion: The effect of explicitly prescribing gait structure is not uniform across methods. For Qwen-STL, the multi-gait formulation substantially extends the range of successful locomotion: Qwen-GA fails from 1.6, m/s onward, while Qwen-MG maintains successful command tracking across the complete evaluation range. GPT-STL and Text2Reward show the opposite trend: their gait-agnostic policies succeed across the complete range, whereas the multi-gait version fails at the high-speed commands. These results suggest that adding gait-specific structure can improve high-speed locomotion, but its benefit depends on the particular reward or specification produced by the language model.

TABLE I
GAIT-AGNOSTIC LOCOMOTION RESULTS ACROSS COMMANDED FORWARD VELOCITIES. LOWER CoT IS BETTER; HIGHER SURVIVAL AND SUCCESS ARE BETTER. EACH ENTRY REPORTS MEAN $\pm$ STANDARD DEVIATION OVER 20 ROLLOUTS.

v_x (m/s)	CoT $\downarrow$	Survival $\uparrow$	Success $\uparrow$
Heuristic			
0.3	1.20 $\pm$ 0.00	100%	100%
0.5	1.00 $\pm$ 0.00	100%	100%
0.7	1.00 $\pm$ 0.00	100%	100%
1.0	1.20 $\pm$ 0.00	100%	100%
1.3	1.30 $\pm$ 0.00	100%	100%
1.6	1.40 $\pm$ 0.00	100%	100%
1.9	1.40 $\pm$ 0.00	100%	100%
2.0	1.40 $\pm$ 0.00	100%	5%
2.1	1.40 $\pm$ 0.00	100%	0%
GPT-GA			
0.3	1.16 $\pm$ 0.01	100%	100%
0.5	1.02 $\pm$ 0.01	100%	100%
0.7	0.99 $\pm$ 0.01	100%	100%
1.0	0.95 $\pm$ 0.01	100%	100%
1.3	0.95 $\pm$ 0.01	100%	100%
1.6	0.95 $\pm$ 0.01	100%	100%
1.9	0.94 $\pm$ 0.01	100%	100%
2.0	0.95 $\pm$ 0.01	100%	100%
2.1	0.97 $\pm$ 0.01	100%	100%
Qwen-GA			
0.3	1.39 $\pm$ 0.02	100%	100%
0.5	1.17 $\pm$ 0.02	100%	100%
0.7	1.10 $\pm$ 0.02	100%	100%
1.0	0.94 $\pm$ 0.01	100%	100%
1.3	0.96 $\pm$ 0.01	100%	100%
1.6	1.04 $\pm$ 0.12	0%	0%
1.9	1.13 $\pm$ 0.07	0%	0%
2.0	1.12 $\pm$ 0.12	0%	0%
2.1	1.19 $\pm$ 0.12	0%	0%
Text2Reward-GA			
0.3	0.91 $\pm$ 0.01	100%	100%
0.5	0.80 $\pm$ 0.00	100%	100%
0.7	0.75 $\pm$ 0.00	100%	100%
1.0	0.74 $\pm$ 0.00	100%	100%
1.3	0.79 $\pm$ 0.00	100%	100%
1.6	0.87 $\pm$ 0.00	100%	100%
1.9	1.00 $\pm$ 0.01	100%	100%
2.0	1.07 $\pm$ 0.01	100%	100%
2.1	1.14 $\pm$ 0.01	100%	100%

Effect of robustness horizon: To understand the impact of the STL robustness evaluation horizon on policy learning, we conduct an ablation study over the window size $H \in \{1, 5, 10, 20, 30\}$ for both gait-agnostic and multi-gait formulations (Tables III and IV). Across most configurations, shorter temporal horizons ($H \in \{1, 5\}$) yield significantly more stable and successful locomotion policies. As the trailing window expands to $H = 20$ and $H = 30$, performance generally degrades. In the gait-agnostic setting, GPT-GA maintains a perfect 100% success rate and optimal cost of transport ($\text{CoT} \leq 1.00$) for $H \leq 10$, but command tracking success collapses to 44% at $H = 20$. Similarly, in the multi-gait setting, Qwen-MG achieves perfect command tracking and high efficiency (CoT of 1.28) at $H = 1$, yet its performance degrades as the window size increases, dropping to a 27.8% success rate at $H = 30$. While GPT-MG exhibits a minor improvement in success rate at longer horizons in the multi-gait task, its overall performance remains well below Qwen-MG’s peak. Furthermore, even when success and survival rates

remain completely invariant to the horizon size, as seen with Qwen-GA, energy efficiency consistently worsens at longer horizons. These results show that a longer robustness horizon is not always beneficial. Because the robustness of $\mathbf{G}$ (always) specifications depends on the worst value in the window, larger H can keep past violations in the reward longer and make credit assignment harder, whereas short horizons provide a more immediate learning signal.

VI. CONCLUSION

We presented an LLM-assisted formal reward-design pipeline for quadruped locomotion. Instead of asking a language model to directly choose reward code and numerical coefficients, GPT-5.5 and Qwen 3.6 propose parametric STL structures; expert trajectories determine their parameters and filter out systematically violated suggestions; and smooth STL robustness provides the PPO reward. The resulting framework supports both gait-aware multi-gait control and a gait-agnostic setting in which locomotion style is allowed to emerge. These comparisons allow us to study what is gained and what remains difficult when temporal logic is inserted between language-model generation and policy learning.

A notable limitation is that our evaluations are currently confined to simulation (MJX). Despite employing domain randomization over friction and actuation parameters, the empirical sim-to-real transferability of these learned multi-gait behaviors has yet to be established on a physical quadruped.

REFERENCES

- [1] P. Dayan and B. W. Balleine, "Reward, motivation, and reinforcement learning," *Neuron*, vol. 36, no. 2, pp. 285–298, 2002.
- [2] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *arXiv preprint arXiv:1804.10332*, 2018.
- [3] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.
- [4] J. Eschmann, "Reward function design in reinforcement learning," *Reinforcement learning algorithms: Analysis and Applications*, pp. 25–33, 2021.
- [5] R. T. Icarte, T. Q. Klassen, R. Valenzano, and S. A. McIlraith, "Reward machines: Exploiting reward function structure in reinforcement learning," *Journal of Artificial Intelligence Research*, vol. 73, pp. 173–208, 2022.
- [6] K. Caluwaerts, A. Iscen, J. C. Kew, W. Yu, T. Zhang, D. Freeman, K.-H. Lee, L. Lee, S. Saliceti, V. Zhuang *et al.*, "Barkour: Benchmarking animal-level agility with quadruped robots," *arXiv preprint arXiv:2305.14654*, 2023.
- [7] K. Zakka, B. Tabanpour, Q. Liao, M. Haiderbhai, S. Holt, J. Y. Luo, A. Allshire, E. Frey, K. Sreenath, L. A. Kahrs *et al.*, "Mujoco playground," *arXiv preprint arXiv:2502.08844*, 2025.
- [8] J. Hare, "Dealing with sparse rewards in reinforcement learning," *arXiv preprint arXiv:1910.09281*, 2019.
- [9] G. Kim, Y.-H. Lee, and H.-W. Park, "A learning framework for diverse legged robot locomotion using barrier-based style rewards," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 10 004–10 010.
- [10] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," *arXiv preprint arXiv:1710.11248*, 2017.
- [11] A. Y. Ng, S. Russell *et al.*, "Algorithms for inverse reinforcement learning," in *ICML*, vol. 1, no. 2, 2000, p. 2.
- [12] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, p. 103500, 2021.

TABLE II

MULTI-GAIT LOCOMOTION RESULTS ACROSS COMMANDED FORWARD VELOCITIES. LOWER CoT IS BETTER; HIGHER SURVIVAL, COMMAND SUCCESS, AND GAIT SUCCESS ARE BETTER. GAIT SUCCESS MEASURES WHETHER THE DESIRED GAIT ASSOCIATED WITH THE COMMANDED SPEED REGIME IS SUCCESSFULLY EXHIBITED.

v_x	CoT ↓	Survival ↑	Success ↑	Gait Match ↑
Expert Oracle				
0.3	0.92 ± 0.02	100%	100%	Success
0.5	0.90 ± 0.01	100%	100%	Success
0.7	0.96 ± 0.01	100%	100%	Success
1.0	1.08 ± 0.01	100%	100%	Success
1.3	1.27 ± 0.01	100%	100%	Success
1.6	1.34 ± 0.02	100%	100%	Success
1.9	1.33 ± 0.01	95%	0%	NA
2.0	1.35 ± 0.01	95%	0%	NA
2.1	1.36 ± 0.01	100%	0%	NA
Human-STL				
0.3	2.10 ± 0.10	100%	100%	Success
0.5	1.50 ± 0.00	100%	100%	Success
0.7	1.20 ± 0.00	100%	100%	Success
1.0	1.20 ± 0.00	100%	100%	Success
1.3	1.10 ± 0.00	100%	100%	Success
1.6	1.00 ± 0.00	100%	100%	Success
1.9	1.10 ± 0.00	100%	100%	Success
2.0	1.10 ± 0.00	100%	100%	Success
2.1	1.10 ± 0.00	100%	100%	Success
GPT-MG				
0.3	1.96 ± 0.04	100%	100%	Success
0.5	1.62 ± 0.03	100%	100%	Success
0.7	1.45 ± 0.03	100%	100%	Success
1.0	1.27 ± 0.02	100%	100%	Success
1.3	1.17 ± 0.01	100%	100%	Success
1.6	1.26 ± 0.01	100%	100%	Partially
1.9	1.27 ± 0.01	100%	0%	NA
2.0	1.28 ± 0.01	100%	0%	NA
2.1	1.28 ± 0.02	100%	0%	NA
Qwen-MG				
0.3	1.97 ± 0.05	100%	100%	Fail
0.5	1.42 ± 0.03	100%	100%	Fail
0.7	1.22 ± 0.02	100%	100%	Fail
1.0	1.18 ± 0.01	100%	100%	Fail
1.3	1.10 ± 0.01	100%	100%	Fail
1.6	1.09 ± 0.01	100%	100%	Success
1.9	1.15 ± 0.02	100%	100%	Success
2.0	1.19 ± 0.01	100%	100%	Success
2.1	1.22 ± 0.01	100%	100%	Success
Text2Reward-MG				
0.3	1.10 ± 0.02	100%	100%	Success
0.5	0.82 ± 0.01	100%	100%	Success
0.7	0.83 ± 0.01	100%	100%	Success
1.0	1.04 ± 0.01	100%	100%	Fail
1.3	1.01 ± 0.01	100%	100%	Fail
1.6	0.98 ± 0.01	100%	100%	Fail
1.9	1.05 ± 0.15	0%	0%	NA
2.0	1.06 ± 0.18	0%	0%	NA
2.1	1.05 ± 0.17	0%	0%	NA

- [13] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems*, vol. 30, 2017.
- [14] D. Youm, H. Jung, H. Kim, J. Hwangbo, H.-W. Park, and S. Ha, "Imitating and finetuning model predictive control for robust and symmetric quadrupedal locomotion," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7799–7806, 2023.
- [15] T. Li, J. Won, J. Cho, S. Ha, and A. Rai, "Fastmimic: Model-based motion imitation for agile, diverse and generalizable quadrupedal locomotion," *Robotics*, vol. 12, no. 3, p. 90, 2023.
- [16] M. Hasanbeig, D. Kroening, and A. Abate, "Deep reinforcement learning with temporal logics," in *International Conference on Formal Modeling and Analysis of Timed Systems*. Springer, 2020, pp. 1–22.
- [17] X. Li, C.-I. Vasile, and C. Belta, "Reinforcement learning with temporal

TABLE III
ABLATION OF STL ROBUSTNESS WINDOW SIZE H FOR GAIT-AGNOSTIC
POLICIES. METRICS ARE AVERAGED ACROSS ALL EVALUATED
COMMANDED FORWARD VELOCITIES.

Method	H	CoT ↓	Survival ↑	Success ↑
GPT-GA	1	0.99	100%	100%
	5	0.99	100%	100%
	10	1.00	100%	100%
	20	1.15	100%	44%
	30	1.14	100%	67%
Qwen-GA	1	1.11	55.6%	55.6%
	5	1.11	55.6%	55.6%
	10	1.26	55.6%	55.6%
	20	1.29	55.6%	55.6%
	30	1.55	55.6%	55.6%

TABLE IV
ABLATION OF STL ROBUSTNESS WINDOW SIZE H FOR MULTI-GAIT
POLICIES. METRICS ARE AVERAGED ACROSS ALL EVALUATED
COMMANDED FORWARD VELOCITIES.

Method	H	CoT ↓	Survival ↑	Success ↑
GPT-MG	1	1.40	100%	55.6%
	5	1.60	100%	55.6%
	10	1.37	100%	55.6%
	20	1.40	100%	66.7%
	30	1.36	99.4%	66.7%
Qwen-MG	1	1.28	100%	100%
	5	1.32	100%	77.8%
	10	2.08	88.3%	60.6%
	20	14.25	77.8%	54.4%
	30	3.59	77.8%	27.8%

logic rewards,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3834–3839.

- [18] H.-C. Liao, “A survey of reinforcement learning with temporal logic rewards,” 2020.
- [19] X. Li, C.-I. Vasile, and C. Belta, “Reinforcement learning with temporal logic rewards,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3834–3839.
- [20] P. Kapoor, A. Balakrishnan, and J. V. Deshmukh, “Model-based reinforcement learning from signal temporal logic specifications,” *arXiv preprint arXiv:2011.04950*, 2020.
- [21] A. Puranic, J. Deshmukh, and S. Nikolaidis, “Learning from demonstrations using signal temporal logic,” in *Conference on Robot Learning*. PMLR, 2021, pp. 2228–2242.
- [22] S. Feng, X. Xinjilefu, W. Huang, and C. G. Atkeson, “3d walking based on online optimization,” in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 2013, pp. 21–27.
- [23] Y. Zhao, U. Topcu, and L. Sentsis, “High-level planner synthesis for whole-body locomotion in unstructured environments,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 6557–6564.
- [24] H. Audren, A. Kheddar, and P. Gergondet, “Stability polygons reshaping and morphing for smooth multi-contact transitions and force control of humanoid robots,” in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 1037–1044.
- [25] Z. Gu, R. Guo, W. Yates, Y. Chen, Y. Zhao, and Y. Zhao, “Walking-by-logic: Signal temporal logic-guided model predictive control for bipedal locomotion resilient to external perturbations,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 1121–1127.
- [26] Z. Gu, Y. Zhao, Y. Chen, R. Guo, J. K. Leestma, G. S. Sawicki, and Y. Zhao, “Robust-locomotion-by-logic: Perturbation-resilient bipedal locomotion via signal temporal logic guided model predictive control,” *IEEE Transactions on Robotics*, 2025.
- [27] J. Humphreys and C. Zhou, “Learning to adapt through bio-inspired

gait strategies for versatile quadruped locomotion,” *Nature Machine Intelligence*, vol. 7, no. 7, pp. 1141–1153, 2025.

- [28] D. DeFazio, Y. Hayamizu, and S. Zhang, “Learning quadruped locomotion policies using logical rules,” in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 34, 2024, pp. 142–150.
- [29] O. Maler and D. Nickovic, “Monitoring temporal properties of continuous signals,” in *International symposium on formal techniques in real-time and fault-tolerant systems*. Springer, 2004, pp. 152–166.
- [30] A. Donzé and O. Maler, “Robust satisfaction of temporal logic over real-valued signals,” in *International conference on formal modeling and analysis of timed systems*. Springer, 2010, pp. 92–106.
- [31] G. E. Fainekos and G. J. Pappas, “Robustness of temporal logic specifications for continuous-time signals,” *Theoretical Computer Science*, vol. 410, no. 42, pp. 4262–4291, 2009.
- [32] A. Balakrishnan and J. V. Deshmukh, “Structured reward shaping using signal temporal logic specifications. in 2019 IEEE/RSJ IROS, 3481–3486,” 2019.
- [33] D. Aksaray, A. Jones, Z. Kong, M. Schwager, and C. Belta, “Q-learning for robust satisfaction of signal temporal logic specifications,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 6565–6570.
- [34] M. Atasever, C. Bakirci, A. R. Corona, K. Azbijari, and J. V. Deshmukh, “Learning gait-aware quadruped locomotion with temporal logic specifications,” *arXiv preprint arXiv:2607.00442*, 2026.
- [35] T. Xie, S. Zhao, C. Wu, Y. Liu, Q. Luo, V. Zhong, Y. Yang, and T. Yu, “Text2reward: Reward shaping with language models for reinforcement learning,” in *International Conference on Learning Representations*, vol. 2024, 2024, pp. 35 663–35 699.
- [36] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, J. Fan *et al.*, “Eureka: Human-level reward design via coding large language models,” in *International conference on learning Representations*, vol. 2024, 2024, pp. 26 516–26 560.
- [37] Y. Chen, R. Gandhi, Y. Zhang, and C. Fan, “Nl2tl: Transforming natural languages to temporal logics using large language models,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 15 880–15 903.
- [38] OpenAI, “GPT-5.5: Large language model,” <https://openai.com>, 2026, accessed: [2026-08-19].
- [39] Qwen Team, “Qwen3.6: Open foundation model,” <https://huggingface.co/Qwen>, Alibaba Cloud, 2026, accessed: [2026-08-19].
- [40] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.
- [41] N. Kohl and P. Stone, “Policy gradient reinforcement learning for fast quadrupedal locomotion,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA’04. 2004*, vol. 3. IEEE, 2004, pp. 2619–2624.
- [42] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.
- [43] E. Asarin, A. Donzé, O. Maler, and D. Nickovic, “Parametric identification of temporal properties,” in *International Conference on Runtime Verification*. Springer, 2011, pp. 147–160.
- [44] Y. Gilpin, V. Kurtz, and H. Lin, “A smooth robustness measure of signal temporal logic for symbolic control,” *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 241–246, 2020.
- [45] Z. Fu, A. Kumar, J. Malik, and D. Pathak, “Minimizing energy consumption leads to the emergence of gaits in legged robots,” *arXiv preprint arXiv:2111.01674*, 2021.
- [46] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots, “Fast and efficient locomotion via learned gait transitions,” in *Conference on robot learning*. PMLR, 2022, pp. 773–783.
- [47] R. M. Alexander and A. Jayes, “A dynamic similarity hypothesis for the gaits of quadrupedal mammals,” *Journal of zoology*, vol. 201, no. 1, pp. 135–152, 1983.
- [48] J. Humphreys, J. Li, Y. Wan, H. Gao, and C. Zhou, “Bio-inspired gait transitions for quadruped locomotion,” *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6131–6138, 2023.