MIMICAGENT: LEARNING QUADRUPED SKILLS VIA TEXT-TO-TRAJECTORY GENERATION

Narayanan Palghat Parameswaran[1*], **Lucky Kant Nayak**[1*], **Neehar Peri**[1], **Deva Ramanan**[1]
[1]Carnegie Mellon University

ABSTRACT

We present MimicAgent, a text-to-trajectory generation framework for learning dynamic quadruped skills. Although reward shaping is extensively used when training quadruped policies, navigating the resulting reward landscape is notoriously difficult, requiring hours of "graduate student descent". Eureka attempts to automate reward design with LLMs, but we find that it struggles to generalize across diverse skills and morphologies. Motivated by the success of example-guided RL for humanoids, we revisit skill learning from demonstrations for quadrupeds. Unlike humanoids, which can exploit large-scale motion capture datasets for learning, quadrupeds lack such reference motion data. We make the observation that manually keyframing quadruped reference motions can be more intuitive than reward shaping; in particular, we find that rather coarse and even dynamically-infeasible motions can still be effective reference targets for example-guided RL. However, manual keyframing is still too cumbersome to create large-scale skill libraries. To address this challenge, we propose an LLM-based pipeline that generates kinematically feasible quadruped trajectories for diverse skills. Although these trajectories are not dynamically feasible, we show that they are sufficient to train successful policies. Across all evaluated skills, human raters consistently prefer policies generated by MimicAgent over those produced by Eureka. See our project page for more details.

1 INTRODUCTION

In recent years, learning-based locomotion policies have demonstrated impressive agility and robustness across a wide range of behaviors Mahankali et al. (2024); Margolis & Agrawal (2023) and terrains Lee et al. (2020). However, training such RL policies typically relies on carefully tuned rewards. For complex and highly dynamic skills, designing bespoke reward functions is notoriously difficult and often requires extensive trial-and-error Kumar et al. (2021); Nahrendra et al. (2023); Long et al. (2023). This challenge has motivated recent efforts to eliminate manual reward engineering entirely.

Automating Reward Design with LLMs. Recent work has explored using LLMs to automate reward design Yu et al. (2023); Ma et al. (2023). For example, Eureka leverages coding LLMs to generate and iteratively refine reward functions from natural language task descriptions. While promising, these methods still depend on hand-engineered success metrics to evaluate candidate rewards and often struggle to generalize across diverse skills and morphologies. In practice, generated reward functions are brittle and often don't accurately follow natural language guidance, particularly for underexplored robot embodiments such as wheeled quadrupeds (Table 1).

Learning from Examples. An alternative to reward-centric RL is learning from demonstrations. Example-guided RL has been a key enabler for recent progress in humanoid locomotion, where motion capture datasets Mahmood et al. (2019); Allshire et al. (2025); Xu et al. (2025); He et al. (2024); Ji et al. (2024); Yu et al. (2021); Luo et al. (2023); Peng et al. (2018b), or manually annotated keyframes Peng et al. (2018a; 2021); Zhang et al. (2025) provide rich supervision. However, quadrupeds lack such large-scale, publicly available datasets. Recent works Bellegarda et al. (2025); Tirumala et al. (2019) curate libraries of simple quadruped motion patterns like walking, trotting, and galloping. However, such efforts are orders of magnitude smaller and less diverse than comparable humanoid

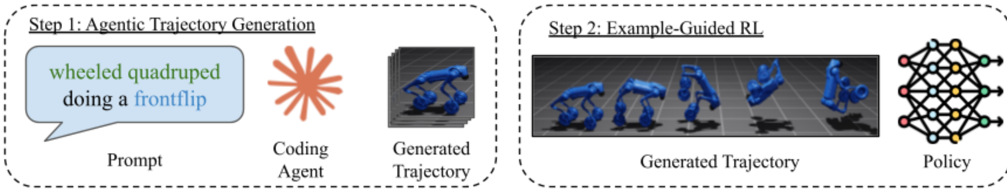


Figure 1: **Overview.** MimicAgent uses LLM coding agents to generate kinematically feasible trajectories from natural language prompts (Step 1). We show that these coarse motions are effective reference targets for example-guided RL policies, and that generating plausible trajectories with an LLM is often easier than designing robust reward functions for the same behaviors (Step 2).

datasets. This data scarcity limits the applicability of demonstration-based methods despite their strong empirical performance.

Agentic Trajectory Generation. To address this gap, we propose MimicAgent, an LLM-based agentic pipeline that directly generates text-guided example trajectories for policy learning. *Our key observation is that it is far easier for a human – and by association, an LLM – to generate reference motions than to generate reward functions.* Given a natural language skill description, MimicAgent produces executable code defining base motion and foot trajectories. This code is then executed in a MuJoCo engine using forward kinematics to produce kinematically feasible trajectories, without running a full physics simulator. Although the resulting trajectories are not dynamically feasible, they provide structured, task-relevant supervision that can be consumed by a downstream imitation-based RL pipeline like DeepMimic Peng et al. (2018a).

Contributions. We present three major contributions. First, we introduce MimicAgent, an LLM-based text-to-trajectory framework for learning dynamic quadruped skills. Although MimicAgent generates dynamically infeasible trajectories, we show that they are sufficient to train successful policies. Our experiments highlight the limitations of prior LLM-based approaches for reward shaping, and demonstrate the effectiveness of using LLMs for trajectory synthesis.

2 Related Works

Quadrupedal Locomotion has matured significantly in recent years Kumar et al. (2021); Lee et al. (2020); Nahrendra et al. (2023); Margolis & Agrawal (2023); Mahankali et al. (2024); Smith et al. (2023). Early work primarily focused on training legged robots to execute complex contact sequences to achieve different locomotion gaits. A common approach is to use model predictive control (MPC) to track predefined contact patterns Grandia et al. (2019); Villarreal et al. (2020), enabling canonical gaits such as trotting Di Carlo et al. (2018), pacing Raibert (1990), bounding Eckert et al. (2015), and galloping Marhefka et al. (2003), as well as non-conventional gaits with customized contact timings Li et al. (2022); Winkler et al. (2018). Despite their success, MPC-based approaches require hand-designed reference trajectories and incur high computational costs, limiting behavioral diversity. Recent work extends these ideas to agile bipedal quadrupedal motions Li et al. (2024) and whole-body loco-manipulation Qiu et al. (2025); Ouyang et al. (2025). In contrast, our approach leverages example-guided reinforcement learning, which can be more intuitive than designing foot contact sequences.

Text-to-Robot Control is an active area of research for both quadrupeds Tang et al. (2023); Zhou et al. (2025); Jian et al. (2025) and humanoids Cui et al. (2024); Tessler et al. (2023); Cui et al. (2025); Tirinzoni et al. (2025). Early methods relied on structured text templates or NLP tools like parse trees to extract constraints and generate robot trajectories Kress-Gazit et al. (2008); Chai et al. (2018); Howard et al. (2014). Other approaches used representation learning to train language-conditioned policies that map free-form instructions directly to actions Stepputtis et al. (2020); Brohan et al. (2022); Mees et al. (2022); Shridhar et al. (2022). However, such approaches require large annotated datasets with paired text and robot control, which is difficult to collect for diverse locomotion behaviors. More recent methods prompt LLMs to generate robot code, bridging the gap between language and motor control via intermediate representations like high-level plans, primitive skills, or trajectories Vemprala et al. (2024); Lin et al. (2023); Liang et al. (2022); Singh et al. (2022); Bucker

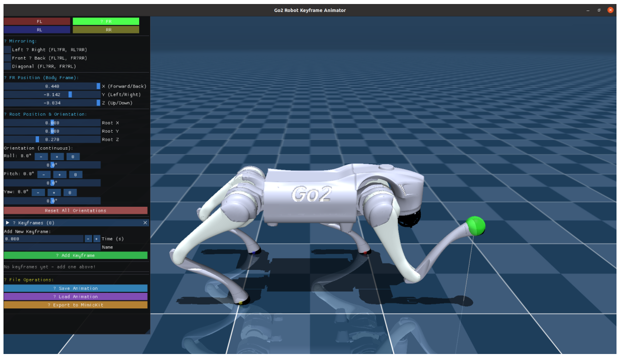


Figure 2: **Keyframing Reference Motion.** We visualize a MuJoCo forward kinematics model for *puppeteering* a quadruped by manually keyframing reference trajectories. We demonstrate that such an interface is easier for both humans and LLMs to tune, compared to standard workflows for reward shaping.

et al. (2022). Inspired by prior work, we leverage LLMs to directly generate reference trajectories for imitation learning.

3 MIMICAGENT: LEARNING DYNAMIC SKILLS WITH LLM-BASED TRAJECTORY SYNTHESIS

MimicAgent aims to automate motion imitation learning by replacing manual reward engineering with an agentic motion synthesis pipeline. Given a natural language description of a quadruped skill (e.g. `wheeled quadruped doing a frontflip`), MimicAgent generates reference motion trajectories that can be directly used by imitation-based RL algorithms. Rather than navigating a complex and brittle reward landscape, MimicAgent leverages the observation that generating a plausible reference trajectory is often significantly easier than designing a robust reward function for the same behavior.

Agentic Motion Synthesis Pipeline. MimicAgent is an iterative pipeline that decomposes motion synthesis into *motion planning, code generation, execution*, and *refinement* (Algorithm 1).

The input to the system is a natural language skill description, which is processed by the motion planner agent to produce a phase-based motion plan encoding the temporal and kinematic structure of the skill over a cyclic phase variable. Each motion plan describes the overall motion intent and stability strategy, the decomposition into sub-phases with associated leg contact states, coordination patterns across leg groups, base linear and angular velocity commands in the body frame, and qualitative leg trajectories relative to the body. It also provides handoff notes for the code generation agent for controller implementation.

Each skill is represented over a normalized phase variable $\phi \in [0, 1]$, allowing motions of arbitrary duration to be expressed in a unified, phase-indexed form. A trajectory is defined as a sequence of keyframes sampled at discrete phase values. Each keyframe specifies the base position $\mathbf{p}(t)$, base orientation $\mathbf{q}(t)$, and joint rotations $\boldsymbol{\theta}(t)$.

$$\mathbf{p}_t \in \mathbb{R}^3, \quad \mathbf{q}_t \in \mathbb{R}^4, \quad \|\mathbf{q}_t\| = 1, \quad \boldsymbol{\theta}_t \in \mathbb{R}^n$$

$$\mathbf{X}_{0:T} = \left\{ \mathbf{x}_t = \begin{bmatrix} \mathbf{p}_t \\ \mathbf{q}_t \\ \boldsymbol{\theta}_t \end{bmatrix} \;\middle|\; t = 0, \ldots, T \right\}$$

This representation enables a compact and interpretable parameterization capable of expressing a wide range of locomotion behaviors.

Code Synthesis and Trajectory Execution. Given the motion plan, a coding agent synthesizes an executable program that defines the temporal functions and sequence for base and foot trajectories. The generated code is executed in a MuJoCo simulator. Importantly, we do not perform any physics simulation. As a result, the synthesized trajectories are kinematically consistent but generally violate dynamic constraints. MimicAgent explicitly embraces this tradeoff, delegating dynamic feasibility to the imitation learning stage.

Algorithm 1 MimicAgent Trajectory Generation

```
def mimic_agent(skill_description, N=4):
    iters = 0
    state = [skill_description]

    motion_plan = motion_planner(skill_description)
    state.append(motion_plan)

    skill_code = coder_agent(motion_plan, state)
    state.append(skill_code)

    while not pass and iters < N:
        trajectory = code_executor(skill_code)
        pass = unit_test(trajectory)
        state.append(result)

        if pass:
            return trajectory

        llm_feedback = motion_diagnosis(state)
        state.append(llm_feedback)

        skill_code = code_modifier(skill_code, state, human_feedback=None
            )
        iters += 1

    return None
```

Task-Agnostic Unit Tests. To validate generated trajectories without introducing task-specific biases, MimicAgent employs a set of *task-agnostic unit tests*. Rather than optimizing task rewards, MimicAgent filters generated motions using task-agnostic constraints, enabling scalable reference trajectory generation without manually specifying task-specific success criteria. These tests are designed to reject physically implausible or degenerate trajectories while remaining broadly applicable across skills. Each unit test computes a scalar reward, and the aggregate score determines whether a trajectory passes validation. The unit tests include:

- *Base Height Envelope.* The base height is constrained to remain within a reasonable envelope relative to the ground. Heights below 0.1 meters or those exceeding twice the standing base height are penalized. This test returns a binary reward of $+1$ or -1.
- *Joint Violation Penalty.* Joint limit violations are penalized proportional to their frequency. The reward is computed as $r_{\text{joint}} = -1 \times N_{\text{violated}}$, where N_{violated} denotes the number of joints exceeding their limits.
- *Ground Penetration.* Foot-ground penetration is penalized based on severity. Trajectories in which all four feet penetrate the ground incur a large penalty, while partial penetration results in a smaller penalty. The reward is bounded between $+1$ and -1.
- *Feet Air-Time Constraint.* For non-aerial skills, at least one foot must remain in contact with the ground for half of the motion cycle. This constraint prevents degenerate motions that unrealistically leave the ground for extended durations.

Importantly, these unit tests do not encode task success, performance objectives, or stylistic preferences. They serve purely as structural validity checks, ensuring that generated trajectories are suitable as imitation references. In contrast, Eureka Ma et al. (2023) relies heavily on carefully designed task-specific success functions to generate reward signals. Defining such success criteria is a time-intensive process and is often a significant bottleneck for learning large skill libraries. Motivated by this limitation, MimicAgent adopts a task-agnostic validation strategy that avoids task-specific success criteria entirely.

Iterative Refinement. If a trajectory fails validation, a diagnostic agent analyzes the failure in the context of the full system state, including the skill description, motion plan, generated code, and unit test outcomes. The diagnostic agent produces natural language feedback identifying likely failure modes. A code modification agent then revises the trajectory-generating program accordingly. This process iterates until a trajectory passes all unit tests or a fixed iteration budget is reached. Valid trajectories are returned for downstream imitation learning. The code modification agent can optionally also take human feedback to improve code generation quality.

By shifting the focus from reward optimization to reference trajectory synthesis, MimicAgent avoids the brittle and time-consuming process of manual reward engineering. Unlike LLM-based reward design methods that require repeated policy training and hand-crafted evaluation criteria, MimicAgent generates supervision directly in trajectory space. This makes the framework efficient, interpretable, and naturally extensible to diverse quadruped skills.

4 EXPERIMENTS

In this section, we briefly describe our evaluation metrics, provide an empirical analysis of our baselines, and ablate the annotation time and impact of reference motion quality between human annotated keyframes and our agentic trajectory synthesis pipeline.

Metrics. We quantitatively compare manual keyframing, Eureka Ma et al. (2023), and MimicAgent with a user study. Participants were presented with pairs of textual task descriptions and videos of trained policies corresponding to that task. Participants were asked to rate how well the policy follows the task description on a 5-point Likert scale. Videos were presented in a randomized order to eliminate ordering bias and participants were not informed which method was used to generate each policy. In addition, we compute the mean squared error between the reference trajectories and the trajectories produced by the learned policy for both human annotated keyframes and MimicAgent's keyframes to measure how faithfully the learned policy follows the demonstration. This allows us to measure the quality and dynamic feasibility of the reference motion.

Baselines. We compare MimicAgent with policies trained by Eureka Ma et al. (2023) and human annotated keyframes. Eureka frames reward design as a program synthesis problem: given a textual skill description and access to environment code (e.g., observations, episode termination conditions, and success function), an LLM generates candidate reward functions. Multiple reward functions are sampled per iteration, trained in parallel, and evaluated using predefined success criteria, with an evolutionary search loop selecting and refining the best-performing rewards. Next, we also evaluate DeepMimic Peng et al. (2018a) policies learned from human annotated keyframes. To simplify the annotation process, we created a custom annotation tool with Claude Sonnet 4.5 to control foot positions and root body pose using click-and-drag mechanics for coarse control and sliders for fine-grained control (Figure 2).

Skills. We evaluate the following skills in our user study. Notably, these five skills are chosen because they are particularly easy to keyframe.

- *UniTree Go2 Trot.* A symmetric diagonal gait where front-left and rear-right legs alternate with front-right and rear-left, maintaining continuous ground contact and a stable torso while tracking commanded forward velocity.
- *UniTree Go2 Bounding.* A dynamic gait where the front legs push off together followed by the rear legs, producing a brief aerial phase and pronounced vertical oscillation while maintaining forward propulsion.
- *UniTree Go2-W Front Flip.* An acrobatic maneuver where the robot generates strong forward pitch momentum, enters a fully airborne phase, rotates 360 about its lateral axis, and lands upright with controlled impact.
- *UniTree Go2-W Side Flip.* A lateral acrobatic motion where the robot initiates a rapid roll rotation about its longitudinal axis, becomes fully airborne, completes a full 360 side rotation, and recovers to a stable stance.
- *UniTree Go2-W Aerial Crossover.* While carving forward on its wheels, the robot lifts alternating diagonal leg pairs off the ground, executing rapid mid-air crossover wave motions.

Comparison with State-of-the-Art. We conduct a user study (n = 30) comparing policies generated from human annotated keyframes, Eureka, and MimicAgent. Participants consistently prefer

Table 1: **Comparison to State-of-the-Art.** We conduct a user study (n=30) to compare policies generated by human annotated keyframes, Eureka, and MimicAgent. We find that users prefer MimicAgent policies for Walking, Side Flip, Front Flip, and Aerial Crossover. Users prefer policies generated by human annotated keyframes for bounding, though all policies receive low ratings. Higher is better.

Skill Robot	**Walking** Go2	**Bounding** Go2	**Side Flip** Go2-W	**Front Flip** Go2-W	**Aerial Crossover** Go2-W
Key Frames	2.5	**3.5**	3.6	4.3	2.6
Eureka	1.8	2.8	1.3	1.4	2.2
MimicAgent	**4.4**	2.5	**4.9**	**4.6**	**4.2**

Table 2: **Comparison of Time-to-Annotation.** We evaluate the time to annotation (in minutes) between human annotated keyframes and MimicAgent keyframes. Notably, MimicAgent is consistently faster than human annotated keyframes. Lower is better.

Skill Robot	**Walking** Go2	**Bounding** Go2	**Side Flip** Go2-W	**Front Flip** Go2-W	**Aerial Crossover** Go2-W
Key Frames	2.6	4.5	4.5	4.4	7.6
MimicAgent	**1.1**	**1.2**	**2.9**	**3.2**	**3.6**

MimicAgent policies for Walking (+2.6 vs. Eureka), Side Flip (+3.6 vs. Eureka), Front Flip (+3.2 vs. Eureka), and Aerial Crossover (+2 vs. Eureka). For Bounding, users favor policies generated from human-annotated keyframes, although all approaches received relatively low ratings. Notably, Eureka still relies on carefully specified task-specific success metrics and exhibits limited robustness when scaling to diverse, stylistically different skills. Although MimicAgent does not require task-specific unit tests, we hand-engineer success criteria for Eureka's policies to give it a better chance of succeeding. Despite this, performance degrades for wheeled quadruped skills (UniTree Go2-W), likely due to the increased complexity of the joint space. We provide videos for all policies used in this user study on our project page.

Time-to-Annotation. We compare the time required for human-annotated keyframes and MimicAgent generated keyframes in Table 2. We find that MimicAgent consistently requires less time than manually annotated keyframes. The five skills evaluated are intentionally limited to relatively simple motions (e.g. unidirectional velocity or in-place yaw) and do not include more complex skills that combine these elements, such as asymmetric leg movements coupled with base rotations. Such skills are difficult and time-consuming to keyframe manually. In contrast, learning-based agents are able to generate these types of skills quickly. We posit that as skill complexity increases, human annotation time will grow significantly faster than MimicAgents skill generation time.

Automatic Skill Generation. In addition to manually prompting MimicAgent, we can leverage LLMs to automatically generate large-scale skill libraries. Figure 3 visualizes the success rates of this automated skill generation process. We first prompt an LLM to propose candidate skills, resulting in 111 total skills. Of these, 11 are deemed infeasible, while 78 successfully converge after four iterations. We further show that incorporating non-expert human feedback improves policy performance, increasing the number of successful skills to 96. Notably, this feedback only provides high-level guidance rather than specific code modifications or hyperparameter tuning. We visually inspect all generated trajectories to determine the success rate.

Comparison with Reference Motion. We compare DeepMimic-style policies trained on reference trajectories generated with our manual keyframing tool (133 frames) and MimicAgent (133 frames) in Table 3. Policies trained with MimicAgent reference trajectories consistently achieved lower error across all skills and evaluation metrics, suggesting that the original reference trajectories were more dynamically feasible than human-annotated trajectories. We track errors for root position in the global frame (E_{RP}, m), root rotation (E_{RR}, rad), joint rotation (E_{JR}, rad), joint velocity (E_{JV}, rad s^{-1}), root linear velocity (E_{RLV}, m s^{-1}), and root angular velocity (E_{RAV}, rad s^{-1}) . This improvement is most pronounced for highly dynamic skills such as side flips, front flips, and aerial crossovers, where policies trained on manually keyframed trajectories exhibit significantly larger velocity and

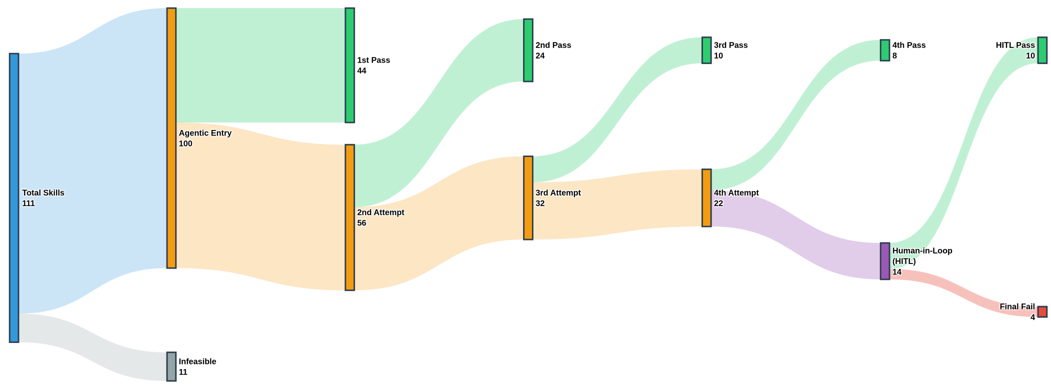


Figure 3: **Automatic Skill Generation.** We use an LLM to propose diverse locomotion skills, yielding 111 candidate skills. We generate trajectories for all skills using MimicAgent: 11 are deemed infeasible, while 78 succeed after up to four iterative attempts. Incorporating human-in-the-loop feedback further improves performance, increasing the total number of successful skills to 96. We visualize several examples in Figure 4.

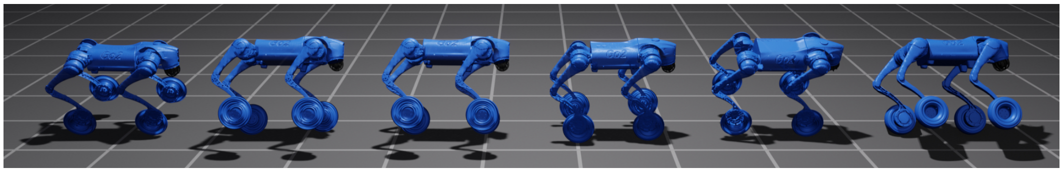
Table 3: **Comparison with Reference Motion.** We evaluate DeepMimic policies trained with human annotated keyframes and MimicAgent keyframes against their respective reference trajectories. Across all skills and metrics, MimicAgent policies achieve lower mean squared error on both pose-level errors (root and body position and rotation), and motion-level errors (linear and angular velocities). Lower is better.

Method Robot	**Error Metric**	**Walking** Go2	**Bounding** Go2	**Side Flip** Go2-W	**Front Flip** Go2-W	**Aerial Cross** Go2-W
KeyFrame Tool	E_{RP}	0.170	0.076	0.199	0.099	0.020
	E_{RR}	0.177	0.208	0.238	0.208	0.025
	E_{JR}	0.360	0.320	0.484	0.507	0.396
	E_{JV}	4.313	3.868	8.380	15.183	6.823
	E_{RLV}	0.251	0.214	0.548	0.287	0.035
	E_{RAV}	0.918	0.517	1.883	1.126	0.233
MimicAgent	E_{RP}	0.052	0.049	0.077	0.089	0.013
	E_{RR}	0.046	0.114	0.207	0.160	0.024
	E_{JR}	0.109	0.224	0.382	0.528	0.373
	E_{JV}	0.646	1.580	5.299	12.871	4.761
	E_{RLV}	0.093	0.136	0.295	0.264	0.044
	E_{RAV}	0.176	0.353	1.869	1.697	0.211

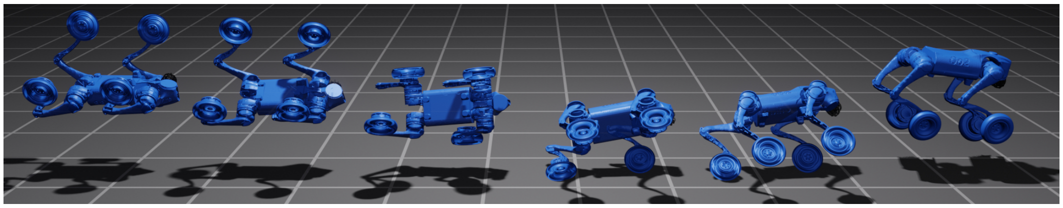
angular velocity errors. In contrast, policies trained on MimicAgent references result in improved motion timing and smoother rotational dynamics during policy learning. Even for simpler skills such as walking and bounding, MimicAgent yields lower global pose error, reflecting higher reference motion quality. These results indicate that improved reference trajectory quality directly reduces downstream imitation error, demonstrating the effectiveness of MimicAgent as a scalable alternative to manual keyframing for imitation-based RL.

Qualitative Results. We visualize policies trained with MimicAgent reference trajectories in Figure 4. We find that MimicAgent can successfully learn policies for diverse text prompts (more videos on the project page).

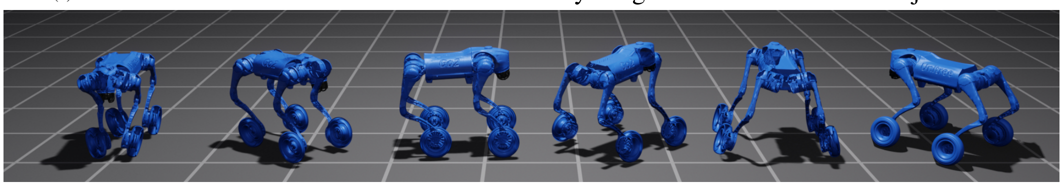
Analysis of Failure Cases. Policies trained for jumping and midair 180° turns failed to learn the intended behaviors. Instead, the robot exploited the task by completing the turn on the ground, likely due to incorrect or near-static foot trajectories that did not sufficiently encourage lift-off. These issues align with recurring failure patterns observed in practice, where agents become trapped in error loops (e.g., alternating between ground penetration and air penalties). This suggests a need for



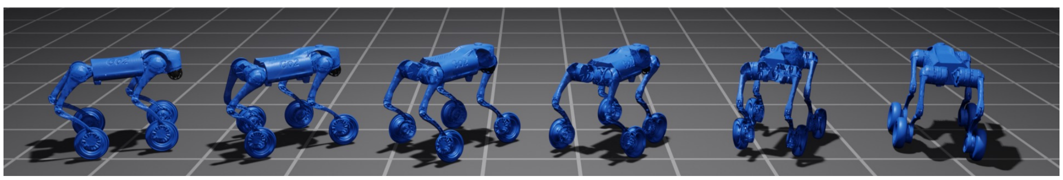
(a) Forward locomotion via large-amplitude lateral base swaying like an inverted pendulum.



(b) Backward lateral locomotion with cartwheel-style leg rotations and circular trajectories.



(c) Legs traces a diamond-shaped pattern on the ground.



(d) Circular locomotion with side-to-side base swaying, like a swinging hammock.

Figure 4: **Qualitative Skill Synthesis Results.** We present representative skills synthesized by MimicAgent across both active and passive motion regimes. Each subfigure shows a different skill (with the initial prompt in the subcaption); please see the project page for trajectory rollouts.

more specialized tool calls and verification checks, such as explicit validation of foot placement and contact conditions.

Limitations and Future Work. Unlike cyclic skills such as walking or skating, more complex, acyclic skills lack direct control over velocity commands or phase durations, making execution and debugging on real hardware difficult. This issue is exacerbated by largely task-agnostic unit tests; while they provide baseline coverage, we observe clear gains from human-in-the-loop feedback. This suggests that skill-specific feedback could further reduce failure rates and improve sim-to-real transfer. Future work should use LLMs to automatically generate skill-specific unit tests and provide targeted feedback during trajectory synthesis.

5 Conclusion

In this paper, we present MimicAgent, an LLM-based agentic pipeline for quadruped trajectory generation. By leveraging coding agents to synthesize coarse yet kinematically feasible reference motions, MimicAgent sidesteps the brittleness and manual effort of reward shaping while avoiding the need for large-scale motion capture datasets. Our results show that, despite being dynamically infeasible, these automatically generated trajectories are sufficient for training robust and expressive behaviors across a diverse set of skills. Human evaluations consistently favor policies trained with MimicAgent over those produced by Eureka, highlighting our method's improved motion quality and language following abilities.

REFERENCES

Arthur Allshire, Hongsuk Choi, Junyi Zhang, David McAllister, Anthony Zhang, Chung Min Kim, Trevor Darrell, Pieter Abbeel, Jitendra Malik, and Angjoo Kanazawa. Visual imitation enables contextual humanoid control. *arXiv preprint arXiv:2505.03729*, 2025.

Guillaume Bellegarda, Milad Shafiee, and Auke Ijspeert. Allgaits: Learning all quadruped gaits and transitions. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15929–15935. IEEE, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Arthur Bucker, Luis Figueredo, Sami Haddadin, Ashish Kapoor, Shuang Ma, Sai Vemprala, and Rogerio Bonatti. Latte: Language trajectory transformer. *arXiv preprint arXiv:2208.02918*, 2022.

Joyce Y. Chai, Qiaozi Gao, Lanbo She, Shaohua Yang, Sari Saba-Sadiya, and Guangyue Xu. Language to action: towards interactive task learning with physical agents. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pp. 29. AAAI Press, 2018. ISBN 9780999241127.

Jieming Cui, Tengyu Liu, Nian Liu, Yaodong Yang, Yixin Zhu, and Siyuan Huang. Anyskill: Learning open-vocabulary physical skill for interactive agents. In *Conference on Computer Vision and Pattern Recognition(CVPR)*, 2024.

Jieming Cui, Tengyu Liu, Ziyu Meng, Jiale Yu, Ran Song, Wei Zhang, Yixin Zhu, and Siyuan Huang. Grove: A generalized reward for learning open-vocabulary physical skill. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15781–15790, 2025.

Jared Di Carlo, Patrick M. Wensing, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Dynamic locomotion in the mit cheetah 3 through convex model-predictive control. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9, 2018. doi: 10.1109/IROS.2018.8594448.

Peter Eckert, Alexander Sprwitz, Hartmut Witte, and Auke Jan Ijspeert. Comparing the effect of different spine and leg designs for a small bounding quadruped robot. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3128–3133, 2015. doi: 10.1109/ICRA.2015.7139629.

Ruben Grandia, Farbod Farshidian, René Ranftl, and Marco Hutter. Feedback mpc for torque-controlled legged robots. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4730–4737. IEEE, 2019.

Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8944–8951. IEEE, 2024.

Thomas M. Howard, Stefanie Tellex, and Nicholas Roy. A natural language planner interface for mobile manipulators. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6652–6659, 2014. doi: 10.1109/ICRA.2014.6907841.

Mazeyu Ji, Xuanbin Peng, Fangchen Liu, Jialong Li, Ge Yang, Xuxin Cheng, and Xiaolong Wang. Exbody2: Advanced expressive humanoid whole-body control. *arXiv preprint arXiv:2412.13196*, 2024.

Pingcheng Jian, Xiao Wei, Yanbaihui Liu, Samuel A Moore, Michael M Zavlanos, and Boyuan Chen. Lapp: Large language model feedback for preference-driven reinforcement learning. *arXiv preprint arXiv:2504.15472*, 2025.

Hadas Kress-Gazit, Georgios E. Fainekos, and George J. Pappas. Translating structured english to robot controllers. *Advanced Robotics*, 22(12):1343–1359, 2008. doi: 10.1163/156855308X344864.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

He Li, Tingnan Zhang, Wenhao Yu, and Patrick M Wensing. Versatile real-time motion synthesis via kino-dynamic mpc with hybrid-systems ddp. *arXiv preprint arXiv:2209.14138*, 2022.

Yunfei Li, Jinhan Li, Wei Fu, and Yi Wu. Learning agile bipedal motions on a quadrupedal robot. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9735–9742. IEEE, 2024.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022.

Kevin Lin, Christopher Agia, Toki Migimatsu, Marco Pavone, and Jeannette Bohg. Text2motion: From natural language instructions to feasible plans. *Autonomous Robots*, 47(8):1345–1365, 2023.

Junfeng Long, Zirui Wang, Quanyi Li, Jiawei Gao, Liu Cao, and Jiangmiao Pang. Hybrid internal model: Learning agile legged locomotion with simulated robot response. *arXiv preprint arXiv:2312.11460*, 2023.

Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. *arXiv preprint arXiv:2310.04582*, 2023.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023.

Srinath Mahankali, Chi-Chang Lee, Gabriel B. Margolis, Zhang-Wei Hong, and Pulkit Agrawal. Maximizing quadruped velocity by minimizing energy. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11467–11473, 2024. doi: 10.1109/ICRA57147.2024.10609983.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pp. 5442–5451, October 2019.

Gabriel B Margolis and Pulkit Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. In *Conference on Robot Learning*, pp. 22–31. PMLR, 2023.

D.W. Marhefka, D.E. Orin, J.P. Schmiedeler, and K.J. Waldron. Intelligent control of quadruped gallops. *IEEE/ASME Transactions on Mechatronics*, 8(4):446–456, 2003. doi: 10.1109/TMECH.2003.820001.

Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters*, 7(4):11205–11212, 2022.

I Nahrendra, Byeongho Yu, and Hyun Myung. Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning. *arXiv preprint arXiv:2301.10602*, 2023.

Yutao Ouyang, Jinhan Li, Yunfei Li, Zhongyu Li, Chao Yu, Koushil Sreenath, and Yi Wu. Long-horizon locomotion and manipulation on a quadrupedal robot with large language models. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11157–11164. IEEE, 2025.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018a.

Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. Sfv: Reinforcement learning of physical skills from videos. *ACM Transactions On Graphics (TOG)*, 37(6):1–14, 2018b.

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 (ToG)*, 40(4):1–20, 2021.

Ri-Zhao Qiu, Yuchen Song, Xuanbin Peng, Sai Aneesh Suryadevara, Ge Yang, Minghuan Liu, Mazeyu Ji, Chengzhe Jia, Ruihan Yang, Xueyan Zou, et al. Wildlma: Long horizon loco-manipulation in the wild. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10011–10019. IEEE, 2025.

Marc H. Raibert. Trotting, pacing and bounding by a quadruped robot. *Journal of Biomechanics*, 23: 79–98, 1990. ISSN 0021-9290. doi: https://doi.org/10.1016/0021-9290(90)90043-3. URL https://www.sciencedirect.com/science/article/pii/0021929090900433. International Society of Biomechanics.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on robot learning*, pp. 894–906. PMLR, 2022.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. *arXiv preprint arXiv:2209.11302*, 2022.

Laura Smith, J Chase Kew, Tianyu Li, Linda Luu, Xue Bin Peng, Sehoon Ha, Jie Tan, and Sergey Levine. Learning and adapting agile locomotion skills by transferring experience. *arXiv preprint arXiv:2304.09834*, 2023.

Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.

Yujin Tang, Wenhao Yu, Jie Tan, Heiga Zen, Aleksandra Faust, and Tatsuya Harada. Saytap: Language to quadrupedal locomotion. *arXiv preprint arXiv:2306.07580*, 2023.

Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. Calm: Conditional adversarial latent models for directable virtual characters. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701597. doi: 10.1145/3588432.3591541. URL https://doi.org/10.1145/3588432.3591541.

Andrea Tirinzoni, Ahmed Touati, Jesse Farebrother, Mateusz Guzek, Anssi Kanervisto, Yingchen Xu, Alessandro Lazaric, and Matteo Pirotta. Zero-shot whole-body humanoid control via behavioral foundation models. *arXiv preprint arXiv:2504.11054*, 2025.

Sashank Tirumala, Aditya Sagi, Kartik Paigwar, Ashish Joglekar, Shalabh Bhatnagar, Ashitava Ghosal, Bharadwaj Amrutur, and Shishir Kolathaya. Gait library synthesis for quadruped robots via augmented random search. *arXiv preprint arXiv:1912.12907*, 2019.

Sai H Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. Chatgpt for robotics: Design principles and model abilities. *Ieee Access*, 12:55682–55696, 2024.

Octavio Villarreal, Victor Barasuol, Patrick M Wensing, Darwin G Caldwell, and Claudio Semini. Mpc-based controller with terrain insight for dynamic legged locomotion. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2436–2442. IEEE, 2020.

Alexander W. Winkler, C. Dario Bellicoso, Marco Hutter, and Jonas Buchli. Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. *IEEE Robotics and Automation Letters*, 3(3):1560–1567, 2018. doi: 10.1109/LRA.2018.2798285.

Sirui Xu, Hung Yu Ling, Yu-Xiong Wang, and Liang-Yan Gui. Intermimic: Towards universal whole-body control for physics-based human-object interactions. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12266–12277, 2025.

Ri Yu, Hwangpil Park, and Jehee Lee. Human dynamics from monocular video with dynamic camera movements. *ACM Trans. Graph.*, 40(6), December 2021. ISSN 0730-0301. doi: 10.1145/3478513.3480504. URL https://doi.org/10.1145/3478513.3480504.

Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, et al. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv:2306.08647*, 2023.

Ziyu Zhang, Sergey Bashkirov, Dun Yang, Michael Taylor, and Xue Bin Peng. Add: Physics-based motion imitation with adversarial differential discriminators. *arXiv preprint arXiv:2505.04961*, 2025.

Kangjie Zhou, Yao Mu, Haoyang Song, Yi Zeng, Pengying Wu, Han Gao, and Chang Liu. Adaptive interactive navigation of quadruped robots using large language models. *arXiv preprint arXiv:2503.22942*, 2025.