

Learning to Recover: Dynamic Reward Shaping with Wheel-Leg Coordination for Fallen Robots

Boyuan Deng^{1,2,*}, Luca Rossini¹, Jin Wang¹, Weijie Wang¹ and Nikolaos Tsagarakis¹

Abstract—Adaptive recovery skills are essential for the practical deployment of wheeled-legged robots, which uniquely combine the agility of legs with the speed of wheels for rapid recovery. However, traditional methods relying on simplified dynamics or sparse rewards often fail to produce robust recovery policies. This paper presents a learning-based framework integrating Episode-based Dynamic Reward Shaping and curriculum learning, which dynamically balances exploration of diverse recovery maneuvers with precise posture refinement. An asymmetric actor-critic architecture accelerates training by leveraging privileged information in simulation, while noise-injected observations enhance robustness against uncertainties. We further demonstrate that synergistic wheel-leg coordination reduces joint torque consumption by 15.85%–26.2% and improves stabilization through energy transfer mechanisms. Extensive evaluations on two distinct platforms achieve recovery success rates up to 99.1% and 97.8% without platform-specific tuning. A video showcase the experimental result is available:

I. INTRODUCTION

Wheeled-legged robots, with their integrated wheel-leg structure, have demonstrated unique mobility advantages in complex environments, establishing them as key platforms for long-duration tasks such as inspection and exploration[1], [2], [3], [15]. However, unexpected falls resulting from dynamic disturbances (e.g., collisions, slippage) often interrupt missions, and existing systems typically rely on manual intervention for recovery, severely limiting operational efficiency and autonomy. Thus, breakthroughs in autonomous post-fall recovery are critical to achieving fully autonomous operations[5]

In essence, recovery after fall is a highly complex motion planning and control problem, necessitating precise posture adjustments through multiple discontinuous contact events[9]. For the controller, deriving an optimal or near-optimal sequence of actions is highly challenging. Owing to the highly nonlinear dynamics of wheeled-legged robots, optimization-based methods often depend on simplified system models and precise state estimation. Consequently, most existing work has focused on pure legged recovery[4], leaving the full potential of wheel-leg collaborative recovery largely unexplored. In contrast, recent advances in reinforcement learning have demonstrated remarkable progress in addressing this issue, offering a new technical pathway toward more flexible and robust recovery strategies. In this

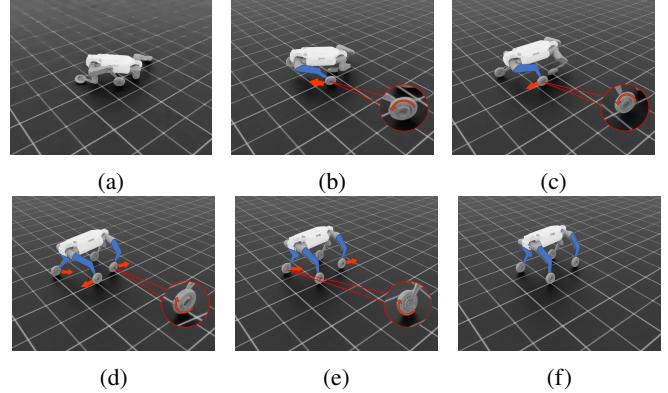


Fig. 1: KYON Robot: Wheel and joint coordination for joint reset and Center-of-Mass adjustment during the recovery process

paper, we explore a widely used learning-based approach to facilitate rapid post-fall recovery for wheeled-legged robots by leveraging their wheels. We employ an asymmetric Proximal policy optimization (PPO) framework, which integrates dynamic exploration of episodes and curriculum learning to train a single network. This design mitigates the sensitivity of recovery and smoothness weight parameters while also avoiding an overly conservative policy tendency. The main contributions of this work are summarized as follows:

- We propose an Episode-based Dynamic Reward Shaping mechanism combined with curriculum learning to balance exploration and policy convergence. This framework enables expansive exploration of diverse recovery maneuvers in early training while progressively refining posture control toward the target configuration, effectively overcoming the limitations of sparse reward designs.
- We systematically validate the critical role of wheel actuation in reducing joint torque consumption and improving stabilization during recovery. This synergy between wheel dynamics and leg adjustments is rigorously quantified through cross-platform experiments, offering new insights into how wheels can actively assist in upright recovery processes.
- Extensive experimental validation on two distinct wheeled-legged platforms (KYON and Unitree Go2-W[11]) demonstrates high adaptability of the proposed approach to varying hardware configurations, achieving success rates up to 97.8% and reduced torque usage without platform-specific tuning, underscoring its robustness and scalability.

*This work was not supported by any organization

¹Humanoids and Human-Centered Mechatronics (HHCM), Istituto Italiano di Tecnologia, Via Morego 30, Genoa, 16163, Italy

²Ph.D. Program of National Interest in Robotics and Intelligent Machines (DRIM), University of Genova, 16126 Genoa, Italy

*Corresponding author: boyuan.deng@iit.it

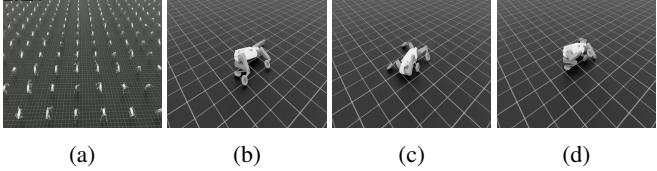


Fig. 2: (a) represents the initialization of the robot's falling during training. (b)-(d) correspond to the initial states of different episodes.

II. RELATED WORK

The post-fall recovery process typically requires a robot to execute a series of actions, gradually transitioning from an initial fallen state to a final standing posture. This process can be formulated as minimizing the discrepancy between the current state and the target state (e.g., base height, joint angles). Due to the highly nonlinear kinematics and dynamics of legged robots, as well as the nonconvex nature of the recovery task itself, some optimization-based methods have resorted to using predefined motion sequences[14], [4] or simplified models[13] to reduce the complexity of optimization while achieving dynamic responsiveness and stable balance. However, these heuristic approaches generally depend heavily on model accuracy and significant manual effort, and they often lack portability across different platforms or tasks.

In recent years, learning-based approaches have shown great promise as alternatives, owing to their model-free nature and the ability to directly learn policies through interaction with the environment. For instance, Lee et al.[7] employ a hierarchical mechanism to decompose the task into two controllers: one for self-righting and one for standing. Hwangbo et al.[6] propose using a single controller to optimize the base height, supplemented with auxiliary rewards to generate more natural recovery motions. Their work also introduces curriculum learning and utilizes a network actuator in simulation to bridge the gap between simulated and real-world performance. More recently, [9] incorporated a manipulator into the recovery process and formulated it as a finite-horizon task, which not only improved recovery success rates but also reduced joint torque consumption.

In reinforcement learning-based methods, relying solely on the final target posture as the primary task reward often leads to sparse feedback, resulting in insufficient exploration and lower success rates. To address this challenge, our study introduces an Episode-based Dynamic Reward Shaping mechanism that balances exploration and convergence. This approach expands the exploration space in the early stages while ensuring that the policy effectively converges to the desired recovery behavior.

III. METHOD

Fig. 2 provides an overview of our training pipeline. The implementation in this work is based on the Isaac Lab framework[10]. In the following, we outline the main components of the training process.

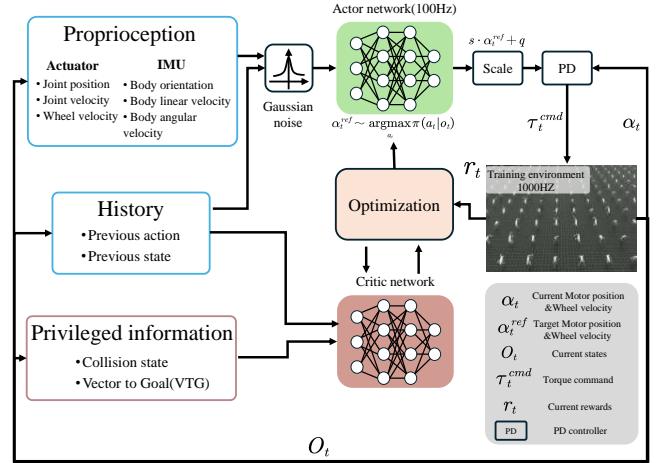


Fig. 3: Framework

A. State initialization and rollout

During the execution of various tasks, wheeled-legged robots may fall or even completely roll over; thus, the recovery controller must accommodate diverse initial conditions. To simulate different fallen states, we randomly initialize the robot's base orientation and joint angles, set the joint torques to zero, and let the robot free-fall from a height of 1.1 m for 2 seconds. This duration is chosen to ensure that the robot fully collapses onto the ground, as shown in the Fig.2, which helps avoid the situation where the dynamics become unrealistic[6]. The diverse set of observed states improves policy generalization and adaptability. Each episode has a fixed duration of 4 seconds and is not terminated early.

B. Privileged information based Actor-Critic

We employ a PPO framework featuring an asymmetric actor-critic architecture, where the critic network leverages privileged information to produce more accurate Q-value estimates and thereby expedite training. The observation inputs for the actor and critic are as follows:

Actor Observation: The policy network observes the robot's state, including current and historical readings (with a time interval of 0.01 s) of the base's linear velocity, angular velocity, orientation, joint positions, joint velocities, and wheel speeds. This temporal information enables the robot to better evaluate its contact state[6]. The actor also receives the previous action taken. Except for the most recent action and joint positions, Gaussian noise is added to the observations to account for state-estimation inaccuracies that arise after a fall. Similar to[5], we apply higher noise levels specifically to joint velocities and the base's linear and angular velocities.

Critic Observation: The privileged information comprises data accessible only in simulation (and not during real-world deployment). By using this information, the critic can estimate Q-values more accurately in the training phase, thereby guiding the actor's actions more effectively. The privileged observations include collision states, which to avoid unrealistic actions that may lead to damaging ground impacts, and VTG information[16] to accelerate learning.

C. Actions

The target values for the joint positions and wheel velocities are computed as $s_p \times a_p + q$ and $s_v \times a_v + q_v$, respectively. Here s_p and s_v are scaling factors, a denotes the action with the highest probability from the policy network's output distribution, and q and q_v represent the default joint angles and default wheel speed. The computed joint position commands are sent to the driver's position PD controller, while the wheel velocity commands are transmitted to the motor speed controller.

In our experiments, we observed that when $s_v > 1.0$, the action values become highly sensitive to changes in velocity control, resulting in noticeable wheel oscillations. Therefore, we chose $s_v < 1.0$ to reduce sensitivity to command variations and enhance motion stability.

D. Dynamic Exploration strategy

Building on an existing reward structure, this study introduces a dynamic episode factor to enhance exploration, thereby extending previous work[5], [9], [6], [16]. Compared with traditional sparse rewards, our approach allows greater deviation in posture during the early stages so that the robot can explore a variety of flipping or rolling maneuvers; it then progressively imposes tighter constraints on the target pose in later stages to guide the policy toward a robust fall-recovery solution. Empirical results show that this method trains a more resilient policy, as demonstrated in the ablation studies in Sec. III-B.

1) *Dynamic Reward Shaping*: In previous research, the post-fall recovery process has often been simplified to a “final pose pursuit” task, typically employing relatively fixed reward or penalty terms based on joint angles, body height, or orientation error. However, in transitioning from a prone position to standing, a wheeled-legged robot may need to undergo large—and sometimes counterintuitive—rolling or swinging movements. Penalizing only the terminal pose too strongly can restrict exploration of diverse state-action pairs, as illustrated in Fig. 2. To balance free exploration in the early stage with precise posture control in the later stage, we propose Episode-based Dynamic Reward Shaping(DS), formally defined as:

$$DS = \left(\frac{a \times t}{T} \right)^k \quad (1)$$

where t is the current step in an episode, $t \in [0, T]$. T is the total number of steps per episode, and k serves as the growth rate, whereas a is the baseline coefficient. For our experiments, we set $a = \frac{T}{2}$, $k = 3$. The immediate reward at each time step is given by:

$$r_t = DS * R(s_t, a_t) \quad (2)$$

Here, $R(s_t, a_t)$ is the original baseline reward function. When the t is close to 0, the agent can attempt substantial posture adjustments, and any successful state-action pairs are retained and reinforced in the policy network. As t approaches the maximum length of the episode, the emphasis is progressively increased on the refinement of the final target

TABLE I: Reward Terms Summary

Reward Term	Definition	Scale
stand joint position	$e^{-\frac{\sum_j (q_j^* - q_j[t])^2}{\sigma_p N_j}}$	42
base height	$e^{-\frac{\max(h^* - h_b[t], 0)^2}{\sigma_h}}$	120
base orientation	$(\mathbf{g}_b - \mathbf{e}_z)^2$	50
body collision	$\sum_{b \in B} \ \lambda_b[t]\ ^2$	-5×10^{-2}
action rate	$\sum_t (a[t] - a[t-1])^2$	-1×10^{-2}
joint velocity	$\sum_j \dot{q}_j^2$	-2×10^{-2}
torques	$\sum_j \tau_j^2$	-2.5×10^{-5}
acceleration	$\sum_j \ddot{q}_j^2$	-2.5×10^{-7}
wheel velocity	$\sum_j \dot{q}_j^2$	-2×10^{-2}

pose. This incremental mechanism promotes action diversity in the early stage of each episode and ensures the stability of the standing posture in the later stage.

We observe that this dynamic reward modulation can slow down learning efficiency. However, as discussed in Sec. III-B, empirical results indicate that it enhances the robustness of the learned policy. This effect can be partly attributed to broader coverage of the state space[17], meaning that more state-action combinations are explored during training, thereby improving the policy’s generalization capability for various fall scenarios.

2) *Curriculum Learning*: To further improve training efficiency and mitigate the reduction in learning speed caused by the dynamic episode factor, we introduce curriculum learning[6], [8], [12]. The curriculum weight (CW) is defined as:

$$CW = CW^\beta \quad (3)$$

where β is the learning rate. We multiply the auxiliary reward by this weight and update it before each environment reset. This approach ensures rapid acquisition of task-related behaviors in the early stages and smoother motion in the later stages, striking a balance between efficiency and feasibility.

E. Reward function

We divide the reward into task rewards and behavior rewards, designed respectively to incentivize correct movements while constraining excessively aggressive behavior. All the robot-state variables required for these computations—such as body acceleration and contact forces—are directly obtained from the simulation environment. The rewards are categorized as follows:

1) *Task Rewards*: To encourage the robot to quickly and accurately achieve the target posture, we define the following three primary sub-rewards:

Base Height: Penalizes deviations between the base height and the desired height to ensure the robot stands upright.

Base Orientation: Penalizes misalignment of the gravitational projection to correct roll and pitch angles.

Joint Position: Penalizes deviations from the default joint angles.

2) *Behavior Rewards*: To encourage smoother recovery, the behavior rewards penalize measures such as the action rate

and joint acceleration. When computing the action rate, the wheel actions are excluded to allow the robot to have greater flexibility in using its wheels for recovery. We also introduce a penalty for body collisions to prevent severe impacts with the ground or self-contact that could damage the robot's structure. Additionally, to stabilize the robot once upright, we provide a reward for the support state, defined as the condition where all four wheels are in contact with the ground simultaneously.

Table I summarizes all reward terms and their scaling. In the reward definitions, q_j , \dot{q}_j and \ddot{q}_j denote the angular position, velocity, and acceleration of each joint. g_b is the gravity vector projected onto the base frame of the robot. B denotes the set of selected robot links, including the shanks, thighs, and base. m_a , a_b , λ_b and F_b represent body mass, body acceleration, contact force, and net force for body b , respectively.

Finally, the overall reward at each time step is defined as:

$$\begin{aligned} r_t = & DS \times r_{joint_pos} + r_{base_height} + r_{base_ori} \\ & + CW \times (r_{collision} + r_{contact_state}) \\ & + DS \times (r_{action_rate} + r_{joint_vel} + r_{torque} + r_{joint_acc})) \\ & + r_{wheelvel} \end{aligned} \quad (4)$$

IV. RESULTS

A. Experimental Setup

All experiments are conducted in the Isaac Sim simulator using KYON and Unitree Go2-W[11] robot models. To enhance policy robustness, extensive domain randomization is applied: the robot's base mass is perturbed by $\pm 5.0\text{kg}$ at the start of each episode, link masses vary by $\pm 10\%$ from their nominal values, and motor strength is randomly scaled within the range [0.8, 1.2]. Moreover, contact dynamics are randomized by uniformly sampling static friction coefficients from [0.7, 1.3] for all robot bodies, while actuator stiffness and damping are perturbed by ± 15.0 and ± 2.0 respectively. Additionally, action noise characterized by a zero mean and a standard deviation of 0.02 is incorporated.

Each episode lasts 4 seconds (at a control rate of 100Hz) and terminates solely upon timeout. The training framework employs an asymmetric PPO algorithm with a learning rate of 0.001 and a discount factor of $\gamma = 0.99$. Furthermore, curriculum learning is integrated, beginning with a difficulty factor of 0.3 that decays exponentially at a rate of 0.968 to progressively enhance policy exploration.

B. Baseline Comparison on Recovery Success Rate

To verify the effectiveness of the Episode-based Dynamic Reward Shaping (DS), we compared the recovery performance of the DS-Policy with that of a baseline under an identical reward function and hyperparameter framework. Recovery was defined as achieving, at the final time step of an episode, a base height exceeding 0.42m , a deviation of the joint angle from the default configuration of no more than 0.5rad , and a maximum joint velocity below 0.1 rad/s . Testing on 2048 randomly generated initial conditions, the

DS-Policy achieved a success rate of 99.1%, representing a 2.7 percentage point improvement over baseline 96.4%.

Further analysis revealed that the failure cases for the baseline were predominantly associated with scenarios in which the front leg joint angles were minimally randomized, shown in Fig4. To illustrate this, we selected a representative initial state from one of the baseline's failure cases and used it as the starting state for the DS-Policy to compare the resulting action sequences. The experiments indicate that, due to the penalty imposed on joint angle deviations, the baseline suppressed front leg motions and attempted to adjust the base height solely through the hind legs, ultimately becoming trapped in a local optimum.

This phenomenon is directly related to differences in the exploration capacity of the action space. Principal Component Analysis (PCA) was applied to the action sequences from the 2048 experiments, and the resulting visualization Fig.5 shows that the variance along the principal components for the DS-Policy ($\sigma_{PC1}^2 = 20.91$, $\sigma_{PC2}^2 = 9.72$) is significantly higher than that for the baseline ($\sigma_{PC1}^2 = 17.33$, $\sigma_{PC2}^2 = 5.22$). These findings confirm that the dynamic reward shaping mechanism, by expanding the exploration boundaries and generating less conservative action sequences, enhances recovery robustness in complex scenarios.

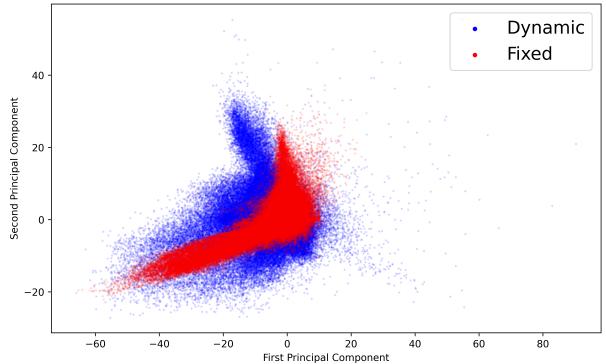


Fig. 5: PCA-based comparison of single-episode action distributions for DS-policy and baseline-policy across 2048 environments.

C. Wheel-Assisted Recovery

In order to evaluate the dynamic contribution of wheeled actuation during recovery, we compared the performance differences between a wheel-leg collaborative mode and a pure joint-driven mode. In the experimental setup, the pure joint-driven group was configured by modifying the robot description file to change the wheel joints from continuously actuated to fixed constraints, and by removing the wheel speed penalty term to eliminate policy bias; all other hyperparameters and reward functions were kept identical to those of the wheel-leg collaborative group. Testing across 2048 random initial states demonstrated that the wheel-leg collaborative mode significantly enhanced recovery stability and optimized joint torque distribution. As shown in Fig.6, although both strategies' base height trajectories converged

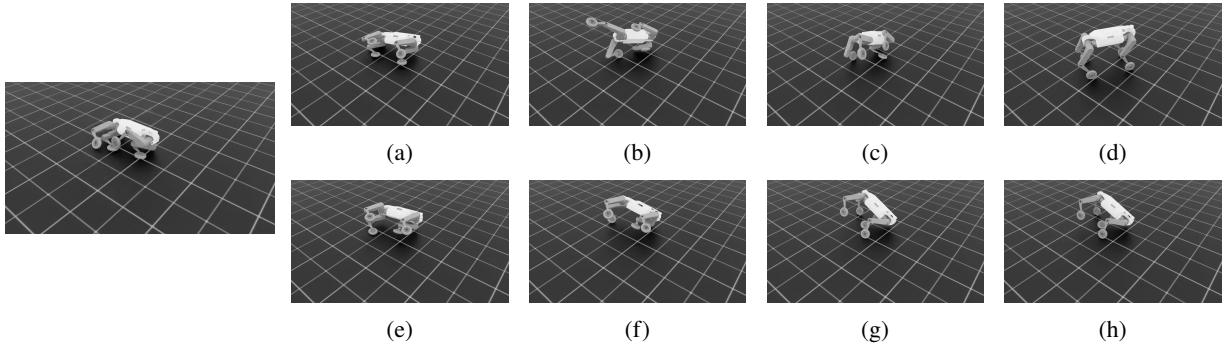


Fig. 4: Recovery processes under the same initial posture using two different strategies. (a)–(d) shows the DS strategy successfully recovery. (e)–(h) illustrates the baseline strategy, which adjusts joint positions and base height but fails to optimize base orientation, becoming trapped in a local optimum and not fully recovering.

to the target height ($0.42m$) within $0.7s$, the wheel-leg collaborative group exhibited lower variance during both the recovery phase ($t < 0.7s$) and the stabilization phase ($0.7s < t < 1.5s$). In contrast, the pure joint-driven group showed larger fluctuations during the stabilization phase, with a variance of 0.102 at $t = 1s$.

Further analysis of joint torque characteristics shown in Fig.8 revealed that the average joint torque in the wheel-leg collaborative group was $35.776N\cdot m$, representing a 15.85% reduction compared to the pure joint-driven group ($42.515N\cdot m$). This reduction is attributed to the conversion of wheel kinetic energy into joint potential energy via wheeled rolling, which decreases the energy consumption required by the joints to counteract gravity, thereby validating the role of wheel assistance in recovery control.

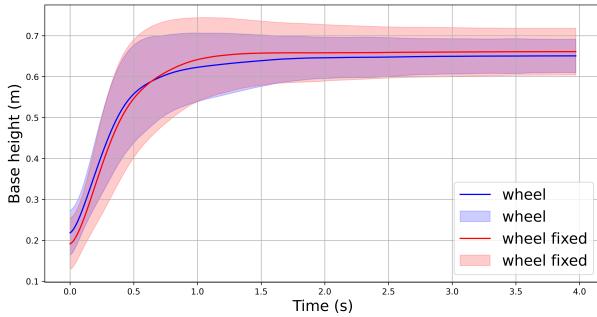


Fig. 6: Comparison of base height dynamics, quantified by mean and variance, for pure joint-driven and wheel-leg coordinated modes over a single episode across 2048 environments.

D. Cross-Platform Validation on Different Robots

To validate the cross-platform generalization capability of the DS strategy, we deployed it on the Unitree Go2-W[11] platform with significantly different configuration parameters and compared its performance with that of the baseline method. The experimental results indicate that despite hardware differences such as mass distribution and joint drive limits, the DS strategy maintained robust recovery performance and demonstrated the advantages of the wheel-leg collaboration mechanism. As illustrated in Fig9, with

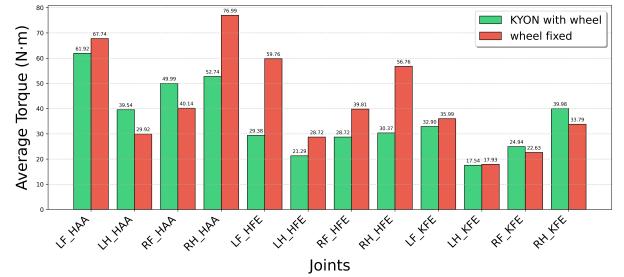


Fig. 7: Comparison of average joint torque for KYON under pure joint-driven and wheel-leg coordinated modes across 2048 environments in a single episode.

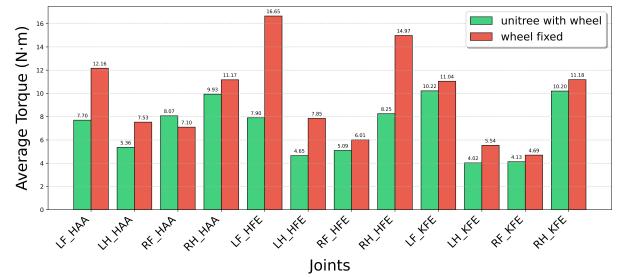


Fig. 8: Comparison of average joint torque for Unitree Go2-W[11] under pure joint-driven and wheel-leg coordinated modes across 2048 environments in a single episode.

the wheels actuated, the Go2-W robot utilized active wheel rotation to buffer its posture, thereby significantly reducing the magnitude of torque adjustments required at the hip joints. Its recovery success rate improved by 3.7 percentage points compared to the baseline strategy, see TableII, and the average joint torque was reduced by 26.2% relative to the fixed-wheel configuration (see TableIII).

These findings are consistent with the experimental results on the KYON platform, indicating that the DS strategy consistently generates optimized action sequences tailored to local degrees of freedom via dynamic reward shaping, and that the wheeled assistance contact force regulation mechanism effectively reduces joint loads.

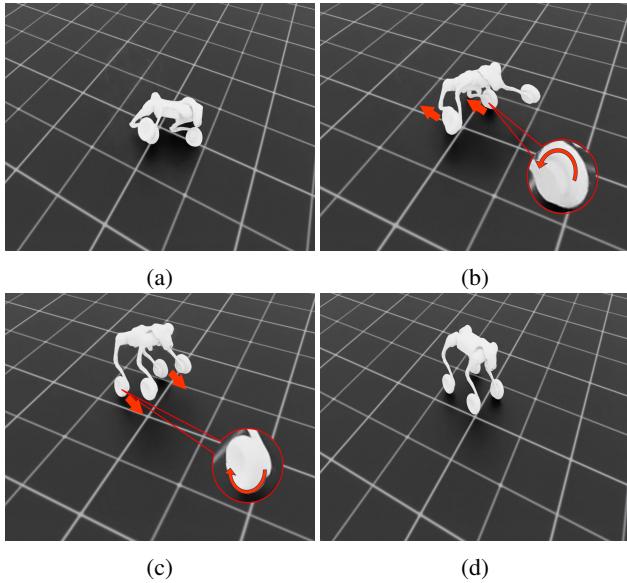


Fig. 9: Key stages of Unitree Go2-W[11]’s recovery process from prone to standing. In (b), after standing up, the robot tilts backward, and backward wheel rotation buffers the tilt to prevent tipping over. Once the center of mass is stabilized in (c), forward wheel rotation quickly adjusts the base height and joint configuration.

TABLE II: Comparison of Recovery Success Rates

Platform	Strategy	Success Rate (%)
KYON	DS	99.1
	Baseline	96.4
Unitree Go2-W[11]	DS	97.8
	Baseline	94.1

TABLE III: Comparison of Average Joint Torque

Platform	Configuration	Avg Torque (N·m)	Reduction (%)
KYON Max Torque:132	Wheel	35.776	
	Wheel Fixed	42.515	15.85
Unitree Go2-W[11] Max Torque:23.7	Wheel	7.123	
	Wheel Fixed	9.657	26.24

V. CONCLUSION

This paper presents a Episode-based Dynamic Reward Shaping for wheel-legged robots to achieve robust and adaptive recovery from fallen states. By integrating dynamic reward shaping and curriculum learning, the proposed method effectively addresses the exploration-convergence dilemma inherent in sparse-reward reinforcement learning. Experimental results demonstrate that DS achieves a recovery success rate of over 97% across diverse platforms, outperforming baseline methods by 3 percentage points. The key innovation lies in the synergy between wheel-assisted motion and leg articulation: wheels provide rapid centroidal adjustment through controlled rolling, while legs enable precise posture stabilization, collectively reducing joint torque demands by 15 – 26% compared to fixed-wheel configurations. Cross-platform validation on robots with distinct kinematic and dynamic parameters (KYON and Unitree Go2-W[11]) further confirms the generalization capability of the framework,

where DS autonomously adapts to hardware-specific constraints without manual tuning. These findings highlight the potential of wheel-leg coordination as a universal strategy for recovery control in hybrid locomotion systems.

Future work will focus on extending the framework to handle more complex terrains (e.g., slopes, gravel) and integrating real-world sensor noise models to bridge the sim-to-real gap. Additionally, the principles of dynamic reward shaping could be applied to other discontinuous-contact tasks, such as multi-robot collaborative recovery or manipulation-oriented posture adjustment.

ACKNOWLEDGMENT

We would like to thank Andrea Patrizi, Despoina Maligiani, Rui Dai, Yifang Zhang Maolin Lei, Jingcheng Jiang, Kuanqi Cai, and Carlo Rizzato for their discussions with me.

REFERENCES

- [1] C Dario Bellicoso, Marko Bjelonic, Lorenz Wellhausen, Kai Holtmann, Fabian Günther, Marco Tranzatto, Peter Fankhauser, and Marco Hutter. Advances in real-world applications for legged robots. *Journal of Field Robotics*, 35(8):1311–1326, 2018.
- [2] Marko Bjelonic, Victor Klemm, Joonho Lee, and Marco Hutter. A survey of wheeled-legged robots. In *Climbing and Walking Robots Conference*, pages 83–94. Springer, 2022.
- [3] Amanda Bouman, Muhammad Fadhil Ginting, Nikhilesh Alatur, Matteo Palieri, David D Fan, Thomas Touma, Torkom Pailevanian, Sung-Kyun Kim, Kyohhei Otsu, Joel Burdick, et al. Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2518–2525. IEEE, 2020.
- [4] Juan Alejandro Castano, Chengxu Zhou, and Nikos Tsagarakis. Design a fall recovery strategy for a wheel-legged quadruped robot using stability feature space. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 41–46. IEEE, 2019.
- [5] Henrik Christensen, Nancy Amato, Holly Yanco, Maja Mataric, Howie Choset, Ann Drobnić, Ken Goldberg, Jessy Grizzle, Gregory Hager, John Hollerbach, et al. A roadmap for us robotics—from internet to robotics 2020 edition. *Foundations and Trends® in Robotics*, 8(4):307–424, 2021.
- [6] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.
- [7] Joonho Lee, Jemin Hwangbo, and Marco Hutter. Robust recovery controller for a quadrupedal robot using deep reinforcement learning. *arXiv preprint arXiv:1901.07517*, 2019.
- [8] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.
- [9] Yuntao Ma, Farbod Farshidian, and Marco Hutter. Learning arm-assisted fall damage reduction and recovery for legged mobile manipulators. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12149–12155. IEEE, 2023.
- [10] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023.
- [11] Unitree Robotics. Unitree go2-w, 2024.
- [12] 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*, pages 91–100. PMLR, 2022.
- [13] Uluc Saranli, Alfred A Rizzi, and Daniel E Koditschek. Model-based dynamic self-righting maneuvers for a hexapedal robot. *The International Journal of Robotics Research*, 23(9):903–918, 2004.

- [14] Claudio Semini, Jake Goldsmith, Bilal Ur Rehman, Marco Frigerio, Victor Barasoul, Michele Focchi, and Darwin G Caldwell. Design overview of the hydraulic quadruped robots. In *The fourteenth Scandinavian international conference on fluid power*, pages 20–22. sn, 2015.
- [15] Marco Tranzatto, Frank Mascarich, Lukas Bernreiter, Carolina Godinho, Marco Camurri, Shehryar Khattak, Tung Dang, Victor Reijgwart, Johannes Loeje, David Wisth, et al. Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge. *arXiv preprint arXiv:2201.07067*, 3, 2022.
- [16] Ted Xiao, Eric Jang, Dmitry Kalashnikov, Sergey Levine, Julian Ibarz, Karol Hausman, and Alexander Herzog. Thinking while moving: Deep reinforcement learning with concurrent control. *arXiv preprint arXiv:2004.06089*, 2020.
- [17] Chong Zhang, Wanming Yu, and Zhibin Li. Accessibility-based clustering for efficient learning of locomotion skills. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1600–1606. IEEE, 2022.