

# HiFAR: Multi-Stage Curriculum Learning for High-Dynamics Humanoid Fall Recovery

Penghui Chen<sup>1</sup>, Yushi Wang<sup>1</sup>, Changsheng Luo<sup>1</sup>, Wenhan Cai<sup>2</sup>, and Mingguo Zhao<sup>1,3</sup>

**Abstract**—Humanoid robots encounter considerable difficulties in autonomously recovering from falls, especially within dynamic and unstructured environments. Conventional control methodologies are often inadequate in addressing the complexities associated with high-dimensional dynamics and the contact-rich nature of fall recovery. Meanwhile, reinforcement learning techniques are hindered by issues related to sparse rewards, intricate collision scenarios, and discrepancies between simulation and real-world applications. In this study, we introduce a multi-stage curriculum learning framework, termed HiFAR. This framework employs a staged learning approach that progressively incorporates increasingly complex and high-dimensional recovery tasks, thereby facilitating the robot’s acquisition of efficient and stable fall recovery strategies. Furthermore, it enables the robot to adapt its policy to effectively manage real-world fall incidents. We assess the efficacy of the proposed method using a real humanoid robot, showcasing its capability to autonomously recover from a diverse range of falls with high success rates, rapid recovery times, robustness, and generalization.

## I. INTRODUCTION

As robots become increasingly integrated into human-centric spaces, ensuring their resilience and adaptability is essential. Given that falls are inevitable in dynamic environments, the autonomous fall recovery capability of humanoid robots is crucial for real-world deployment.

The need for rapid fall recovery in humanoid robots is driven by the dynamic and unpredictable nature of real-world environments. Quick recovery from falls minimizes downtime and potential damage, enhancing the robot’s operational efficiency and safety. For instance, in competitive environments like RoboCup [1], quick and efficient fall recovery is crucial to maintain performance.

Early solutions for fall recovery relied on finite state machines (FSMs) to trigger preprogrammed actions upon detecting fall indicators [2], [3]. While computationally lightweight—enabling real-time execution on early humanoids like Honda’s ASIMO [4], their brittleness became evident in unstructured environments.

The need for real-time adaptation in dynamic environments spurred the adoption of optimization-based methods, particularly for maintaining bipedal stability under external perturbations. A seminal advancement in this domain is the Capture Point theory [5], which formulates push recovery as a problem of regulating the robot’s zero-moment point

(ZMP) within its support polygon. By solving convex optimization problems, methods like Model Predictive Control (MPC) [6] enabled humanoids to recover balance through step adjustment and torso reorientation. In the context of robot fall recovery, optimization-based methods, such as Whole-Body Control (WBC) [7], have proven effective in online optimization of the robot’s fall trajectory [8]. However, unmodeled actuator dynamics lead to trajectory tracking errors, and computational latency becomes intractable for multi-contact scenarios.

Model-based control approaches, while effective for stable locomotion, struggle to address the high-dimensional dynamics and contact-rich nature of fall recovery due to their reliance on precise system modeling and computational latency. Reinforcement learning (RL) has been proven effective in humanoid locomotion [9] and emerges as a solution for fall recovery by enabling robots to learn recovery policies through environment interaction rather than explicit physics modeling. Recent advances in RL have shown success in developing long-horizon behaviors [10], whole-body control [11], and contact-rich tasks [12], [13], making it a promising approach for fall recovery.

Various approaches have been proposed to solve fall recovery with reinforcement learning, including predefined trajectories [10], [14], symmetric behaviors [15], key state initialization(KSI) [16], and task stage division [17]. However, these methods have limitations in adaptability to different fall scenarios or robustness to external disturbances.

Autonomous fall recovery in dynamic environments is a critical challenge for humanoid robots. This work aims to develop a robust policy that enables a fallen humanoid robot to autonomously recover and stand up across various fall scenarios, including supine, prone, and lateral falls. To systematically evaluate the proposed fall recovery policy, we employ the following criteria:

- **Success Rate:** The percentage of successful recovery trials.
- **Recovery Time:** The time required to recover from a fall.
- **Robustness:** The ability to recover from falls under external disturbances and uncertainties.
- **Generalization:** The ability to recover from falls under different initial states and environmental conditions.

To address these challenges, we introduce a multi-stage curriculum learning framework that progressively refines the fall recovery policy by increasing task dimensionality and complexity. In the initial stage, the policy is trained in a low-dimensional setting to establish fundamental recovery

\*This research was funded by Beijing Municipal Natural Science Foundation grant number 24L30170

<sup>1</sup>Department of Automation, Tsinghua University, Beijing 100084, China

<sup>2</sup>Booster Robotics Technology Co., Ltd, Beijing, China

<sup>3</sup>Beijing Innovation Center for Future Chips, Tsinghua University, Beijing 100084, China

behaviors. Subsequently, the second stage extends to a high-dimensional deployment scenario, incorporating additional constraints and variability. Each stage integrates tailored optimization techniques to enhance policy robustness and stability.

Our key contributions include:

- We introduce a stage division strategy and curriculum setting to break down the complex high-dimensional fall recovery task into simpler low-dimensional tasks and gradually increase the task complexity to facilitate the learning process.
- We employ KSI and reward shaping to guide the learning process and accelerate the convergence of a stable fall recovery policy. Furthermore, we introduce dimensionality expansion through supplementary actuated joints, enabling the policy to generalize robustly across diverse fall scenarios.
- We conduct comprehensive experimental validation on the real humanoid robot Booster T1, which illustrates the versatility and robustness of the proposed approach. Real robot experiment videos are available on our project page<sup>1</sup>.

## II. RELATED WORKS

### A. Learning Real-world Humanoid Fall Recovery

Tuomas Harnoja *et al.* trained a DRL agent with pre-defined keyframes on a kidsize humanoid[10]. However, forcing the robot to follow certain trajectories limited the fall recovery behaviors.

FRASA [15] integrates fall recovery and stand-up strategies for the Sigmaban humanoid [18] into a unified framework but simplifies the problem to a planar scenario and enforces symmetric behaviors, which limits the agent’s capabilities.

By using a KSI method, Chuanyu Yang *et al.* proposed a DRL framework that can learn versatile fall recovery policies for different humanoid and quadruped robots in simulated environments. However, this framework has only been validated on the real Jueying Pro quadruped robot [16].

Ziwen Zhuang *et al.* achieved humanoid standing through mobile manipulation guided by high-level motion commands, yet their approach lacks autonomous recovery capabilities. [14].

Furthermore, Tao Huang *et al.* proposed the HoST [17] method to train fall recovery by segmenting long-horizon motions into multiple short-horizon sub-tasks. However, this explicit segmentation may limit the agent’s adaptability to diverse fall scenarios, such as prone and lateral falls.

HumanUp [19] also utilizes a two-stage curriculum learning framework to train humanoid robot stand-up. However, their second stage is a tracking task of a slowed-down trajectory discovered in the first stage, which limits the agent’s recovery speed and adaptability to different fall scenarios.

### B. Multi-Stage Curriculum Learning

Multi-Stage Curriculum Learning (MSCL) represents a training methodology that incrementally escalates the complexity of learning tasks by introducing supplementary constraints or objectives at each phase [20]. Empirical evidence suggests that multi-stage learning and curriculum learning enhances the sample efficiency and generalization capabilities of reinforcement learning agent [21], [22], particularly within complex and high-dimensional environments such as fall recovery training [23].

### C. State Initialization

Reference state initialization (RSI) [24] samples the initial state of the agent from the reference motion at the start of each episode. KSI [16] initializes the agent’s state to a predefined key state, while do not require a reference motion sequence.

RSI leverages the rich and informative state distribution of reference motion to guide the agent during training. KSI contributes to the stabilization of the learning process and enhances the convergence of the policy by offering the agent multiple consistent starting points.

### D. Reinforcement Learning Framework

Many open-source reinforcement learning frameworks are available for humanoid robots, such as Legged Gym and Humanoid Gym [25]. We leverage Booster Gym [26] as the foundational codebase to implement our methods. It incorporates pre-configured settings and techniques for sim-to-real transfer, along with a comprehensive toolchain supporting policy evaluation across multiple simulators and real-world deployment. This framework collectively reduces engineering overhead while accelerating the development of our methods.

## III. METHODS

### A. Framework and Curriculum Setup

The high-dimensional nature of the fall recovery task, coupled with the complexities of collisions and contacts, as well as the sparse reward structure, presents significant challenges in **the development of an effective fall recovery policy**. The variability of fall scenarios encountered in real-world applications, along with hardware constraints and the discrepancies between simulated environments and actual conditions, further complicate the learning and implementation of such policies, thereby hindering **the training of a deployable fall recovery strategy**. Consequently, the direct application of single-stage reinforcement learning to develop an autonomous and robust fall recovery policy proves to be problematic.

To mitigate these challenges, we propose a multi-stage curriculum learning framework, referred to as HiFAR, as illustrated in Fig. 1. This framework incrementally introduces the agent to increasingly complex and high-dimensional recovery tasks. The multi-stage curriculum learning framework is divided into two distinct stages:

<sup>1</sup><https://hi-far.github.io/>

## Training Phase 1

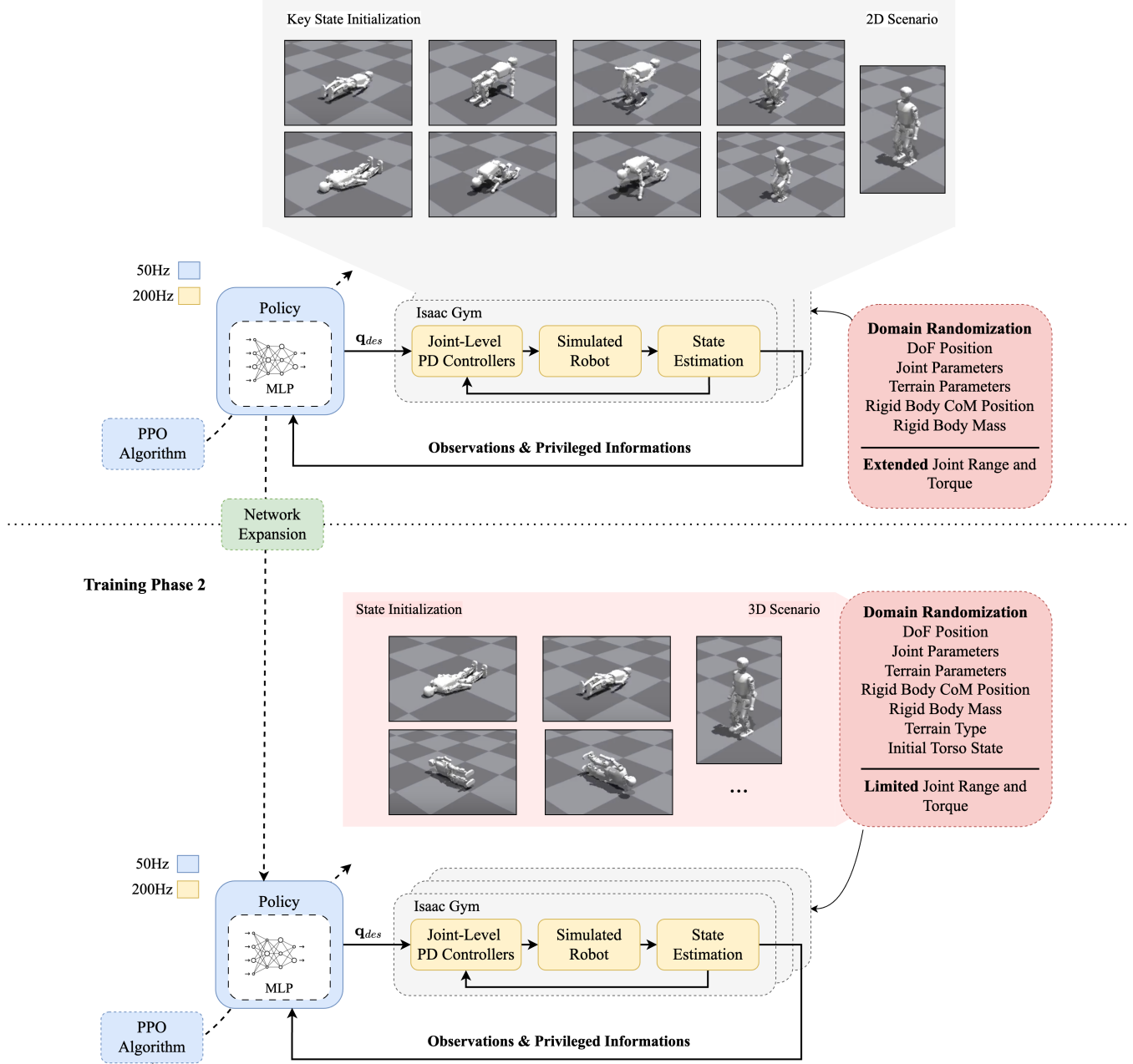


Fig. 1. **Framework overview.** HiFAR offers a multi-stage curriculum learning framework designed for autonomous fall recovery in humanoid robots. The initial training phase emphasizes the development of a fall recovery policy within a low-dimensional task environment. The subsequent training phase tackles the intricacies associated with formulating a deployable fall recovery policy in a higher-dimensional task setting.

The initial stage concentrates on **the development of a basic fall recovery policy**. In this phase, the task is constrained to two-dimensional fall scenarios (*e.g.*, supine and prone falls), with control limited to joints operating within the  $(x, z)$  plane (the joints highlighted in yellow in Fig. 2(A)). This design choice is based on the observation that humanoid robots can effectively recover from these standard positions using predominantly planar movements. By constraining the policy’s exploration space, this configuration reduces the risk of self-collision during training. The agent is trained to

recover from simple falls by utilizing an extended range of motion and higher torque capacities. Techniques such as KSI and reward shaping are employed to facilitate the learning process.

The subsequent stage focuses on **the training of a deployable fall recovery policy**. This phase expands the task to encompass three-dimensional fall scenarios, such as cross-leg falls. To handle diverse real-world falls, we include lateral hip roll joints (orange in Fig. 2(A)) in the action space and introduce further randomizations in terrain and robot state.

Realistic constraints on joint positions, torque, and velocity ensure practical training.

The curriculum across the two stages includes the following components:

- **Task Dimension:** The initial stage is centered on a two-dimensional task, whereas the subsequent stage progresses to a three-dimensional task.
- **Task Constraints:** The first stage is characterized by lenient constraints that promote exploration, in contrast to the second stage, which imposes stringent constraints and introduces complex perturbations.
- **Task Complexity:** The initial stage is concerned with straightforward fall scenarios, while the second stage incorporates a greater variety of randomized fall scenarios.

By first mastering simpler tasks, the agent is better prepared to tackle more complex challenges, ultimately resulting in more efficient and stable strategies for fall recovery.

### B. Training Setup

The agent is trained in a simulated environment using the Booster Gym framework [26]. The target state of fall recovery is the robot standing up with a stable posture, defined by target CoM height, DoF positions, and torso orientation, as shown in Fig. 2(B).

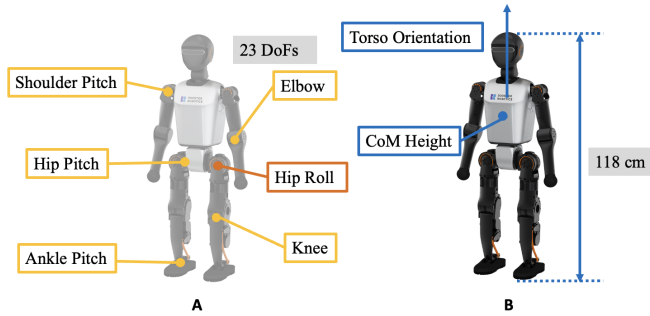


Fig. 2. **Booster T1.** (A) Controlled joints. (B) Target Standing State.

The agent has proprioceptive observations, including the robot’s orientation, angular velocity, joint positions, velocities, and action. The action space consists of the target joint position of controlled joints. We use the Proximal Policy Optimization (PPO) algorithm [27] and RMA network architecture [9] to train the agent.

We achieve cross-stage network expansion by integrating zero-initialized fully-connected layers into both input and output layers of the pre-existing network, thereby enabling the second stage to possess expanded action decision spaces and enhanced environmental perception dimensions while preserving knowledge transfer from the initial training phase.

Instead of forcing symmetric actions like FRASA[15], we use a mirror loss term to encourage the agent to learn symmetric recovery strategies while allowing for asymmetry when necessary, which is more realistic and generalizable to real-world scenarios.

### C. Reference State Initialization

As illustrated in Figure 1, we have selected six critical states—comprising three prone keyframes and three supine keyframes—from established handcrafted fall recovery motions. These motions are specifically designed to facilitate recovery from both supine and prone falls and have demonstrated efficacy in practical applications. The identified key states are used to initialize the agent’s state at the beginning of each episode, ensuring a consistent starting point for the learning process. This approach stabilizes the training process and improves policy convergence.

### D. Reward Design

Table I outlines the reward function components, designed to encourage prompt and stable fall recovery. Each component is weighted by a coefficient  $\omega_i$  to balance competing objectives, with adjustments across learning stages for task alignment.

Survival, base height, standing, orientation, and DoF reference terms facilitate recovery and upright stability. Conversely, torque, DoF velocity/acceleration, root acceleration, action rate, DoF limits, torque fatigue, power, and hip roll terms penalize undesirable actions, promoting smooth, efficient recovery.

TABLE I  
COMPONENTS OF THE REWARD FUNCTION

Reward terms	Formulas
Survival	$R_{\text{survival}} = 1$
Base Height	$R_{\text{base height}} = \exp(-\sigma(h_{\text{base}} - h_{\text{des}})^2)$
Standing	$R_{\text{stand}} = 1_{\text{standing}}$
Orientation	$R_{\text{orientation}} = \ \mathbf{g}_{x,y}\ ^2$
DoF Reference	$R_{\text{dof pos ref}} = \ \mathbf{q}_{\text{ref}} - \mathbf{q}\ ^2$
Torques	$R_{\text{torques}} = \ \boldsymbol{\tau}\ ^2$
DoF Velocities	$R_{\text{dof vel}} = \ \dot{\mathbf{q}}\ ^2$
DoF Accelerations	$R_{\text{dof acc}} = \ \ddot{\mathbf{q}}\ ^2$
Root Accelerations	$R_{\text{root acc}} = \ \mathbf{a}\ ^2$
Action Rate	$R_{\text{action rate}} = \ \dot{\mathbf{a}}_t\ ^2 + \ \ddot{\mathbf{a}}_t\ ^2$
DoF Position Limits	$R_{\text{dof pos limits}} = \ \mathbf{1}_{\text{out of bound}}\ $
Torque Fatigue	$R_{\text{torque fatigue}} = \ \boldsymbol{\tau} \oslash \boldsymbol{\tau}_{\text{limit}}\ ^2$
Power	$R_{\text{power}} = \ \boldsymbol{\tau} \odot \dot{\mathbf{q}}\ $
Hip Roll	$R_{\text{hip roll}} = (q_{\text{hip roll}} - q_{\text{ref}})^2$

### E. Domain Randomization

We apply domain randomization techniques, including randomized push forces and torques, within the simulation environment to improve the robustness and generalizability of the learned policy. The randomized parameters are detailed in Table II:

Additionally, Booster Gym [26] facilitates the modeling of control delays within the simulation, thereby increasing

TABLE II  
DOMAIN RANDOMIZATION PARAMETERS

Parameter	Range	Operation	Distribution
DoF position	[0., 0.05]	Additive	Gaussian
Base XY Position	[-1., 1.]	Additive	Uniform
Base Linear Velocity	[0., 0.1]	Additive	Gaussian
Joint Stiffness	[0.95, 1.05]	Scaling	Uniform
Joint Damping	[0.95, 1.05]	Scaling	Uniform
Terrain Friction	[0.1, 2.0]	Additive	Uniform
Terrain Compliance	[0.5, 1.5]	Additive	Uniform
Terrain Restitution	[0.1, 0.9]	Additive	Uniform
Torso CoM Position	[-0.1, 0.1]	Additive	Uniform
Torso Mass	[0.8, 1.2]	Scaling	Uniform
Other CoM Position	[-0.005, 0.005]	Additive	Uniform
Other Mass	[0.98, 1.02]	Scaling	Uniform

the realism of the training process and enhancing the applicability of the learned policy to real-world scenarios.

#### IV. RESULTS AND ANALYSIS

##### A. Training Analysis

We analyze the agent’s learning curve with KSI. As shown in Fig. 3, the standing reward curve exhibits four step-like increases, indicating that the agent progressively learns to recover from falls and stand up at each key state. The integration of KSI with selected key states stabilizes the learning process and enhances policy convergence.

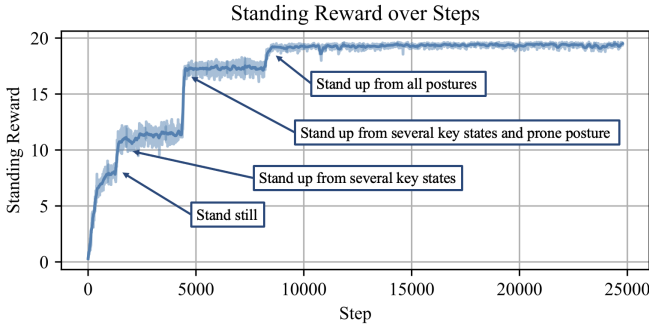


Fig. 3. **Training Analysis.** Standing reward function curve and agent behavior during training.

##### B. Simulation Experiments

1) *Simulation Settings:* We evaluate our policy in the Webots simulator based on success rate, recovery time, and robustness. The simulation environment is designed to replicate real-world conditions, including the robot model and control system.

Before deploying the policy on the physical robot, we conduct a series of simulation experiments to assess its effectiveness in fall recovery. The standard experimental

setup includes supine and prone fall scenarios, with a torso mass of 11.7 kg, a friction coefficient of 1.0, and no external disturbances. Each experiment is repeated 10 times under identical conditions to measure the recovery success rate.

2) *Standard Supine and Prone Recovery Experiments:* Snapshots of the simulation experiments are presented in Fig. 4. In both prone and supine scenarios, our policy successfully enables the robot to recover from falls and stand upright with a stable posture, achieving a 100% success rate. The rapid variations in the torso pitch curve highlight the robot’s highly dynamic movements.

3) *Disturbance Experiments:* In this experiment, we apply a random push force to the robot at 0.8s after the recovery process starts and lasts 200ms.

In the planar force experiment, the perturbation force is applied forward along the robot’s torso direction during recovery from the prone position and backward during recovery from the supine position. The recovery success rate under different perturbation forces is presented in Table III.

TABLE III  
SUCCESS RATE OF PLANAR FORCE EXPERIMENTS

Force	50N	100N	150N	200N	250N	300N
Prone	100%	100%	100%	100%	80%	0%
Supine	100%	100%	100%	100%	100%	60%

In the lateral force experiment, the perturbation force is applied leftward along the robot’s torso direction during recovery from both prone and supine positions. The recovery success rate under different perturbation forces is shown in Table IV.

TABLE IV  
SUCCESS RATE OF LATERAL FORCE EXPERIMENTS

Force	50N	100N	150N	200N	250N
Prone	100%	100%	100%	90%	10%
Supine	100%	100%	100%	100%	0%

The results demonstrate the strong robustness of our policy against perturbations in various directions.

4) *Load Experiments:* We simulate the mass inaccuracy of the real robot by modifying its mass in simulation. The success rate of the recovery process under different mass perturbations is shown in Table V.

TABLE V  
SUCCESS RATE OF LOAD EXPERIMENTS

Extra Torso Mass	+2.4kg	5.9kg	+9.4kg	+11.7kg
Prone	100%	100%	100%	100%
Supine	100%	100%	100%	0%

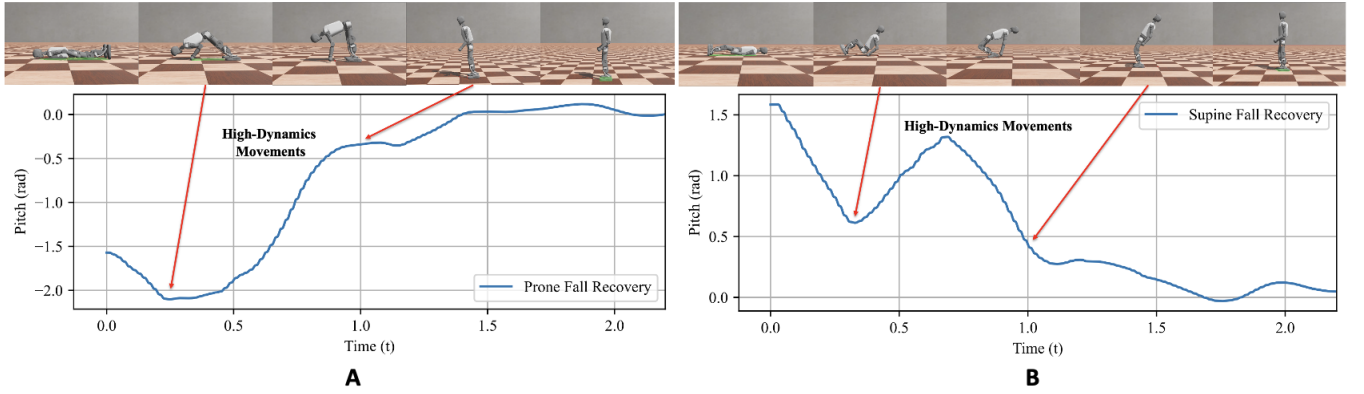


Fig. 4. Snapshots of simulation experiments and the curve of the torso pitch. (A) Prone fall recovery. (B) Supine fall recovery.

With up to 80% extra torso mass, the robot can still recover from prone and supine falls with a 100% success rate, demonstrating the strong robustness of our policy against mass perturbations.

5) *Friction Experiments*: In real-world scenarios, the robot may fall on terrains with varying friction coefficients. To evaluate its ability to recover from falls under different surface conditions, we conduct experiments with various friction coefficients. The recovery success rate for each friction setting is presented in Table VI.

TABLE VI  
SUCCESS RATE OF FRICTION EXPERIMENTS

Friction Coefficient	0.8	0.6	0.4	0.2	0.1
Prone	100%	100%	100%	90%	0%
Supine	100%	100%	100%	100%	90%

The high recovery success rate at friction coefficients as low as 0.2 demonstrates the robustness of our policy and its applicability to real-world terrains with low friction.

6) *Torque Limits Experiments*: In real-world conditions, motors may be unable to generate the desired torque due to operational constraints. To assess the robot's ability to recover from falls under varying torque limitations, we conduct experiments with different torque limits. The success rate of the recovery process under different torque limits is shown in Table VII.

TABLE VII  
SUCCESS RATE OF TORQUE LIMIT EXPERIMENTS

Torque Limit Percentage	85%	75%
Prone	100%	100%
Supine	100%	60%

The high recovery success rate at an 85% torque limit demonstrates that the robot can reliably recover from falls despite actuator wear and degradation.

## V. REAL-WORLD EXPERIMENTS AND ANALYSIS

### A. Hardware Platform

We validate our policy on the real humanoid robot Booster T1, which is a 118cm tall humanoid robot with 23 DoFs. The robot is equipped with a set of sensors, including IMUs, and joint encoders to provide feedback for the control system. On-board computations are performed using a Nvidia AGX Orin GPU and 14-core high-performance CPU.

### B. Experimental Setup

1) *Supine and Prone Recovery Experiments*: Initially placed in either a supine or prone position, the robot executes the policy to restore itself to an upright posture. We record the success rate and recovery time for each trial.

2) *Random Initial State Experiments*: Initialized in various fall conditions, including lateral falls, crossed-leg falls, and sitting positions, the robot's ability to recover from diverse initial states is assessed.

3) *Environmental Complexity Experiments*: Recovery performance is evaluated in complex scenarios, such as standing up from a slope, recovering with obstacles (e.g., a ball) between its legs, and operating in outdoor environments.

4) *Load and Disturbance Experiments*: The robot's recovery ability is tested under additional loads and subjected to strong external forces, such as pushes or impacts, to evaluate its robustness and stability.

### C. Results and Analysis

1) *Supine and Prone Recovery Experiments*: The performance of our policy is shown in Figure 5. Our policy demonstrates high-dynamics recovery behaviors on the real robot, closely matching its performance in the simulation environment. This consistency validates the sim-to-real transferability of our method.

To evaluate recovery performance, we conducted 20 supine and 20 prone fall recovery trials on the real robot. The average recovery time and success rates are summarized in Table VIII. Our method achieves rapid and reliable fall recovery with a 100% success rate, demonstrating its robustness in real-world deployment.



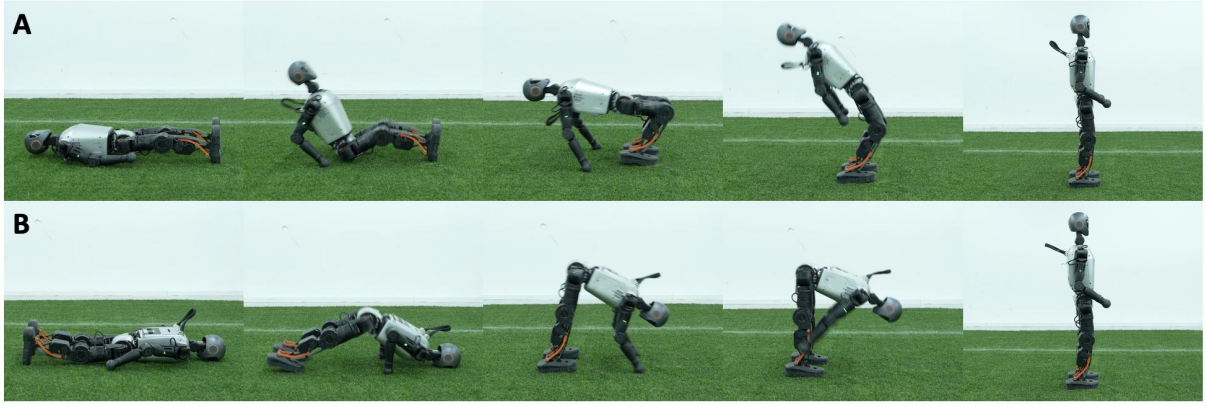


Fig. 5. **Fall recovery behaviors on a real robot.** (A) High-dynamics fall recovery from the prone position. (B) High-dynamics fall recovery from the supine position.

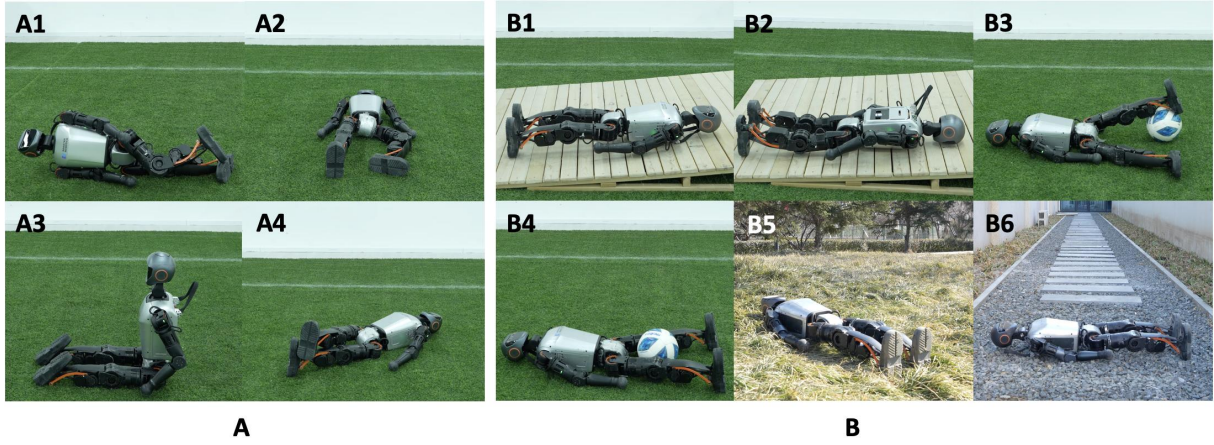


Fig. 6. **Fall recovery under different scenarios.** (A) Random initial state. (A1) lateral positioning, (A2) apart leg posture, (A3) seated position, and (A4) crossed leg posture. (B) Environmental complexity. (B1-2) sloped surfaces, (B3-4) obstacles, and (B5-6) outdoor environments.

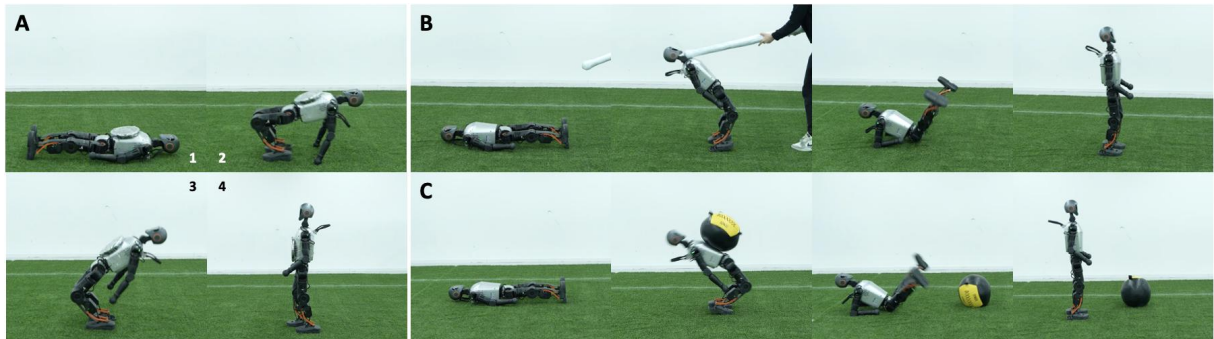


Fig. 7. **Fall recovery under load and disturbance.** (A) 5kg load. (B) Push. (C) Impact.

2) *Random Initial State and Environmental Complexity Experiments:* To assess generalization, we initialize the robot in various random states, as illustrated in Figure 6(A). The robot successfully recovers from all scenarios, demonstrating the adaptability of our policy.

Further, we test the robot in complex environments, including standing up from a slope, navigating obstacles between its legs, and recovering in outdoor conditions (Figure 6(B)). The robot consistently executes versatile recovery be-

haviors across diverse environmental conditions, showcasing the generalizability of our approach.

3) *Load and Disturbance Experiments:* We evaluate recovery performance under additional physical constraints. Figure 7(A) illustrates the robot with a 5 kg load attached during recovery. The robot successfully stands up while maintaining stability, demonstrating its ability to handle additional payloads.

Furthermore, we introduce external disturbances by ap-

TABLE VIII  
AVERAGE TIME AND SUCCESS RATE OF SUPINE AND PRONE RECOVERY  
EXPERIMENTS

	Supine	Prone
Average Time (s)	2.711 $\pm$ 0.319	2.875 $\pm$ 0.474
Success Rate	100%	100%

plying push forces (Figure 7(B)) and impact (Figure 7(C)) during recovery. The robot reacts by adopting a safe sitting posture to mitigate impact before immediately executing high-dynamic recovery to a standing stance. These results confirm the robustness of our policy in handling strong disturbances autonomously.

4) *Various Behaviors: Push and Fall Recovery:* The robot uses ankle and hip strategies for minor perturbations and recovery steps for stronger pushes. If falling, it transitions to a safe posture before recovering. **Walking after Recovery:** After standing up, the robot stabilizes in a standard stance, enabling a seamless transition into locomotion, demonstrating the practicality of our approach.

## VI. CONCLUSIONS

We introduce a multi-stage curriculum learning approach for autonomous humanoid fall recovery. Our method progressively increases task complexity, starting with basic recovery skills and advancing to complex scenarios while addressing sim-to-real challenges.

Simulation and real-world experiments confirm our method's high success rate, fast recovery, robustness, and generalization across diverse falls and disturbances. The approach enables versatile recovery behaviors in varied environments.

This method offers a robust solution for humanoid fall recovery and can extend to other adaptive robotics applications. Future work will aim to enhance the policy's adaptability to a wider range of scenarios, particularly in environments where space or speed constraints affect the robot's stand-up ability.

## ACKNOWLEDGMENT

We thank Booster Robotics for T1 hardware and support.

## REFERENCES

- [1] R. Federation, "Robocup official website," <https://www.robocup.org/>, 2025.
- [2] F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, S. Kajita, K. Yokoi, H. Hirukawa, K. Akachi, and T. Isozumi, "The first humanoid robot that has the same size as a human and that can lie down and get up," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 1633–1639.
- [3] J. Stückler, J. Schwenk, and S. Behnke, "Getting back on two feet: Reliable standing-up routines for a humanoid robot," in *IAS*. Citeseer, 2006, pp. 676–685.
- [4] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent asimo: System overview and integration," in *IEEE/RSJ international conference on intelligent robots and systems*, vol. 3. IEEE, 2002, pp. 2478–2483.
- [5] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *2006 6th IEEE-RAS international conference on humanoid robots*. Ieee, 2006, pp. 200–207.

- [6] S. Katayama, M. Murooka, and Y. Tazaki, "Model predictive control of legged and humanoid robots: models and algorithms," *Advanced Robotics*, vol. 37, no. 5, pp. 298–315, 2023.
- [7] F. L. Moro and L. Sentis, "Whole-body control of humanoid robots," *Humanoid robotics: a reference*, pp. 1161–1183, 2019.
- [8] Z. Cai, Z. Yu, X. Chen, Q. Huang, and A. Kheddar, "Self-protect falling trajectories for humanoids with resilient trunk," *Mechatronics*, vol. 95, p. 103061, 2023.
- [9] A. Kumar, Z. Li, J. Zeng, D. Pathak, K. Sreenath, and J. Malik, "Adapting rapid motor adaptation for bipedal robots," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1161–1168.
- [10] T. Haarnoja, B. Moran, G. Lever, S. H. Huang, D. Tirumala, J. Humpalik, M. Wulfmeier, S. Tunyasuvunakool, N. Y. Siegel, R. Hafner, et al., "Learning agile soccer skills for a bipedal robot with deep reinforcement learning," *Science Robotics*, vol. 9, no. 89, p. eadi8022, 2024.
- [11] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang, "Expressive whole-body control for humanoid robots," *arXiv preprint arXiv:2402.16796*, 2024.
- [12] Q. Zhang, P. Cui, D. Yan, J. Sun, Y. Duan, G. Han, W. Zhao, W. Zhang, Y. Guo, A. Zhang, et al., "Whole-body humanoid robot locomotion with human reference," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 11 225–11 231.
- [13] J. Dao, H. Duan, and A. Fern, "Sim-to-real learning for humanoid box loco-manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 930–16 936.
- [14] Z. Zhuang and H. Zhao, "Embrace collisions: Humanoid shadowing for deployable contact-agnostics motions," *arXiv preprint arXiv:2502.01465*, 2025.
- [15] C. Gaspard, M. Duclaud, G. Passault, M. Daniel, and O. Ly, "Frasa: An end-to-end reinforcement learning agent for fall recovery and stand up of humanoid robots," *arXiv preprint arXiv:2410.08655*, 2024.
- [16] C. Yang, C. Pu, G. Xin, J. Zhang, and Z. Li, "Learning complex motor skills for legged robot fall recovery," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 4307–4314, 2023.
- [17] T. Huang, J. Ren, H. Wang, Z. Wang, Q. Ben, M. Wen, X. Chen, J. Li, and J. Pang, "Learning humanoid standing-up control across diverse postures," *arXiv preprint arXiv:2502.08378*, 2025.
- [18] J. Allali, L. Gondry, L. Hofer, P. Laborde-Zubieta, O. Ly, S. N'Guyen, G. Passault, A. Pirrone, and Q. Rouxel, "Rhuban football club-team description paper," Technical report, Tech. Rep., 2019.
- [19] X. He, R. Dong, Z. Chen, and S. Gupta, "Learning getting-up policies for real-world humanoid robots," *arXiv preprint arXiv:2502.12152*, 2025.
- [20] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.
- [21] D. Wang, C. Liu, F. Chang, H. Huan, and K. Cheng, "Multi-stage reinforcement learning for non-prehensile manipulation," *IEEE Robotics and Automation Letters*, 2024.
- [22] Y. Chen, C. Yang, T. Hu, X. Chen, M. Lan, L. Cai, X. Zhuang, X. Lin, X. Lu, and A. Zhou, "Are u a joke master? pun generation via multi-stage curriculum learning towards a humor llm," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 878–890.
- [23] T. Tao, M. Wilson, R. Gou, and M. Van De Panne, "Learning to get up," in *ACM SIGGRAPH 2022 conference proceedings*, 2022, pp. 1–10.
- [24] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions On Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.
- [25] X. Gu, Y.-J. Wang, and J. Chen, "Humanoid-gym: Reinforcement learning for humanoid robot with zero-shot sim2real transfer," *arXiv preprint arXiv:2404.05695*, 2024.
- [26] B. Robotics, "Booster gym: A user-friendly reinforcement learning framework for humanoid robot," [https://github.com/BoosterRobotics/booster\\_gym](https://github.com/BoosterRobotics/booster_gym), 2024.
- [27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.