2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Mixture-of-Experts Policy for Smooth and Stable Multi-Posture Fall Recovery in Bipedal Robot

Haomin Rong[1], Yuying Chen[1], Zhiyong Xu[1], Lijie Xie[1], Qingyu Yan[2], Hui Cheng[1,*]

***Abstract*— Bipedal robots are inherently prone to falling due to their higher center of mass and narrower support polygon, making automatic fall recovery a long-standing challenge. Existing approaches often rely on posture-specific strategies or exhibit limited robustness and generalization, restricting their real-world applicability. We present a unified Mixture-of-Experts (MoE) framework that trains a single policy capable of recovering from diverse fallen configurations. By leveraging base height estimation and proprioceptive history within a gating mechanism, the framework dynamically allocates recovery tasks to specialized experts, yielding smooth and stable motions. Extensive real-world experiments show that the policy transfers zero-shot to hardware and consistently achieves recovery not only under repeated disturbances, but also from highly challenging postures and even on inclined slopes—demonstrating robustness and generalization beyond prior methods.**

I. INTRODUCTION

In real-world deployment, the inherent risk of falling poses a critical challenge to the operational reliability of bipedal robots. Compared with quadrupedal systems, bipedal robots have a higher center of mass and a narrower support polygon, making them particularly susceptible to losing balance under unexpected disturbances. Recent advances in reinforcement learning (RL) [1], [2] have enabled bipedal robots to perform a wide range of movements across diverse terrains, from walking to complex jumping [2]–[5]. Nonetheless, collisions or external perturbations can still lead to unintended falls, often requiring human intervention or execution of predefined recovery motions. Even optimization- and RL-based recovery strategies that improve success rates typically focus on a single posture, struggling to generalize to diverse initial fall configurations. Additionally, extremely rapid joint motions during recovery may induce instabilities or cause mechanical damage.

Most existing methods [6], [7] allow robots to stand after a fall, but each fall posture generally requires a separate recovery strategy. Predefined controllers demand detailed specification of postures and contact sequences, limiting scalability and adaptability to unseen configurations. Similarly, RL-based recovery policies trained on a single posture exhibit poor performance on novel fall poses, leading to low training efficiency and restricted applicability in real-world scenarios.

1 H.M. Rong, Y.Y. Chen, Z.Y. Xu, L.J. Xie and H. Cheng are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China.
2 Q.Y. Yan is with Orbot Company.
* Corresponding author: chengh9@mail.sysu.edu.cn

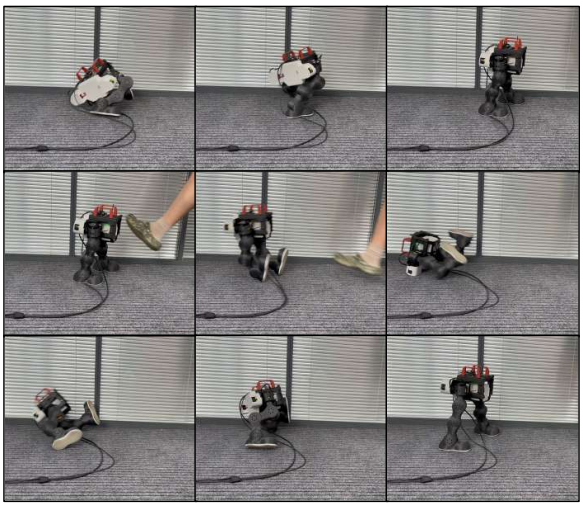

Fig. 1: Snapshots of the robot fall recovery and repeatedly recovering from external unexpected disturbance, demonstrating a robust and smooth recovery process.

This paper introduces a Mixture-of-Experts (MoE) framework [8], [9] for bipedal fall recovery. The proposed method uses single-stage training and dynamically allocates recovery tasks based on the robot's posture and proprioceptive history, including estimated height. A temporally optimized reward regulates standing-up speed to limit excessive joint motion, while an adaptive tracking factor [10] improves post-recovery stability. To our knowledge, this is the first application of MoE to bipedal fall recovery, achieving robust and smooth recovery across diverse postures with direct real-world deployment.

In summary, the contributions of this work are as follows:

- We propose a unified MoE framework where a single policy recovers from diverse fall postures by using height estimation and proprioceptive history to dynamically route tasks to specialized experts, enabling robust recovery and zero-shot deployment on real hardware.
- Smooth and hardware-compatible recovery is achieved through a temporally optimized reward and a velocity-regulation curriculum, which jointly suppress torque spikes, limit peak joint speeds, and enhance robustness against external perturbations.
- Postural stability in the standing phase is reinforced with an adaptive tracking factor that balances position and velocity terms relative to the robot's posture, effectively mitigating oscillations and ensuring stable upright main-

tenance.

The remainder of this paper is organized as follows. Section II introduces related works on MoE locomotion and fall recovery for legged robots. Section III introduces the problem formulation. Section IV describes the MoE-based policy and its training and control methods. Section V exhibits and discusses the simulation and real-world experimental results. Finally, a brief conclusion of this paper is given in Section VI.

II. RELATED WORK

A. Mixture-of-Experts Locomotion

MoE is a machine learning architecture that improves model performance and efficiency by combining multiple specialized expert networks, with a gating network selecting the most suitable expert for a given input. This mechanism allows a single model to handle diverse tasks or input conditions more effectively than monolithic architectures. In locomotion control, MoE has shown advantages in managing complex, multi-modal behaviors. Prior work applies MoE for unified quadrupedal and bipedal gait control, dynamically assigning experts to different terrains to enable agile locomotion [11], employs residual MoE architectures to refine humanoid gait styles while maintaining realism via AMP rewards [12], [13], and coordinates upper- and lower-body control for full-body humanoid teleoperation to overcome unnatural motions [14]. These studies demonstrate that MoE enables more expressive and adaptable policies, a capability leveraged here for robust multi-posture fall recovery in bipedal robots.

B. Legged Robots Fall Recovery

1) Model-based Recovery: Before RL–based control, robotic fall recovery mainly relies on model-based methods with manually designed trajectories or contact-conditioned online searches. Full-body dynamics with linear optimization enable quadrupeds to resist strong disturbances [15], but rely on simplified assumptions. Combining static stability with dynamic joint trajectories facilitates bipedal standing [16], yet trajectory-centric designs are brittle under unmodeled perturbations. Closed-loop, tree-structured retrieval of action sequences improves adaptability [17], but depends on the completeness of predefined libraries. Overall, these methods lack scalability and robustness for diverse, unpredictable falls in real-world settings.

2) Reinforcement Learning Recovery: RL has demonstrated strong potential for robotic fall recovery by enabling complex motor skills and improving robustness in dynamic environments. For quadrupeds, approaches estimate mass distribution and collisions using a Variational Autoencoder (VAE) [18], design morphology-specific dynamic rewards [19], or incorporate predefined contact sequences [20]. However, these methods often rely on accurate modeling assumptions or constrain recovery to specific behaviors, limiting flexibility. In humanoids, two-stage exploration–imitation pipelines [7], curriculum-based force assistance [6], and mirror loss combined with domain randomization [21] facilitate learning, yet they typically require complex training, struggle to generalize across diverse fall postures, and may produce abrupt or unstable motions that hinder smooth recovery.

III. PROBLEM FORMULATION

We model the bipedal fall-recovery task as an infinite-horizon Partially Observable Markov Decision Process (POMDP), defined by $(\mathcal{S}, \mathcal{A}, \mathcal{P}, O, r, \gamma)$. The robot must recover from arbitrary fallen postures to a stable standing state. Here, $\mathcal{S}$ is the underlying state space, $\mathcal{A}$ the continuous joint action space, and O the proprioceptive observations available to the controller. Transitions follow $\mathcal{P}(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$, the reward $r(\mathbf{s}_t, \mathbf{a}_t)$ encourages safe and efficient recovery, and γ balances short- and long-term objectives.

We train policies with an asymmetric actor–critic framework [22], where the actor uses partial observations while the critic exploits privileged simulation states for sample-efficient learning. Optimization is performed with Proximal Policy Optimization (PPO) [23], maximizing the expected discounted return:

$$\pi_\theta^* = \arg\max_\theta \ \mathbb{E}\Big[\sum_{t=0}^{T-1} \gamma^t r_t\Big], \tag{1}$$

where r_t is the reward at time t, $\gamma \in [0, 1)$ the discount factor, and T the episode horizon.

IV. APPROACH

We propose a unified framework that enables bipedal robots to recover from diverse fallen postures with a single policy. The method integrates reinforcement learning with a MoE architecture, where a height estimator guides expert selection, and employs optimized rewards, curriculum training, and adaptive tracking to achieve stable and natural stand-up motions. An overview of the proposed framework is illustrated in Fig. 2.

A. MoE Policy Architecture

We adopt a MoE architecture, where specialized experts handle different recovery scenarios and a gating network blends their outputs based on proprioceptive observations and estimated height. This design enables robust recovery from varied postures and smoother standing motions.

1) Observation Space: Our policy receives the input $\mathbf{o}_t = [h_t, \mathbf{p}_{t-14:t}] \in \mathbb{R}^{541}$, where h_t denotes the estimated base height and $\mathbf{p}_{t-14:t}$ represents a sequence of 15 consecutive proprioceptive measurements. Each entry $\mathbf{p}_t \in \mathbb{R}^{36}$ includes joint positions, joint velocities, previous action, base angular velocity, and projected gravity from the IMU.

2) Height Estimator Network: The MoE gating network is conditioned on the robot's base height to provide task-relevant context. Since base height is not directly observable on hardware, it is treated as privileged information in simulation, and a deployable estimator is learned.

During training, the simulator ground-truth height h_t^{priv} supervises a predictor f_ψ that maps the last 15 timesteps of base positions $\mathbf{p}_{t-14:t}$ to an estimate $\hat{h}_t$:

$$\hat{h}_t = f_\psi(\mathbf{p}_{t-14:t}). \tag{2}$$

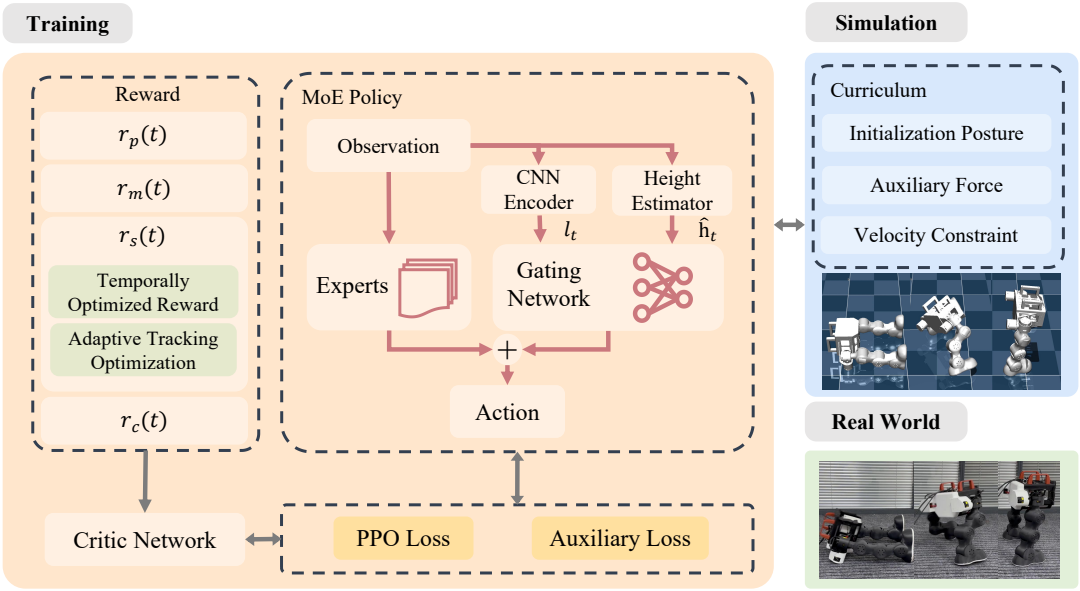

Fig. 2: Overview of our approach. Observations are encoded and fed to the MoE policy, where a height estimator informs the gating network for expert selection. Experts generate actions, with training guided by optimized rewards, curriculum learning, and adaptive tracking across three simulation stages. The resulting policy is deployed on hardware for robust multi-posture recovery.

The estimator is a two-layer multilayer perceptron (MLP) [24] with 256 and 128 ELU-activated units producing a scalar output, trained with mean squared error:

$$\mathcal{L}_{f_\psi} = ||\hat{h}_t - h_t^{priv}||^2, \tag{3}$$

using Adam with learning rate 10^{-3} and mini-batches from simulation. Once trained, f_ψ provides real-time height estimates for the MoE gating network on hardware.

3) Policy Network: We implement the policy network using an MoE architecture, consisting of 6 expert networks and a gating network G_σ. Each expert is implemented as a MLP with three hidden layers of 512, 256, and 128 units, which maps robot state observations to an action distribution.

The gating network determines the contribution of each expert based on the robot's estimated base height $\hat{h}_t$ and a temporal stack of 15 consecutive observations $\mathbf{p}_{t-14:t}$. These stacked observations are processed by a convolutional neural network(CNN) encoder [25], consisting of a 1×1 convolution, a depthwise separable convolution (kernel size 3), adaptive average pooling, and a fully connected layer, producing an 8-dimensional latent vector l_t that captures short-term temporal dependencies and spatial correlations. Conditioning the gating network on both $\hat{h}_t$ and l_t allows it to adaptively select experts according to the robot's current posture and recovery phase. Specifically, lower $\hat{h}_t$ values indicate a fallen state, assigning higher weights to experts specialized in standing-up behaviors, while near-nominal heights shift weights toward experts optimized for stabilization.

Formally, the gating weights are computed as:

$$w_i = \text{softmax}\big(G_\sigma(\hat{h}_t, l_t)\big)[i], \quad i = 1, \ldots, n, \tag{4}$$

where n is the number of experts and $\sum_i w_i = 1$. The final policy action is obtained by a weighted sum of expert outputs:

$$\mathbf{a}_t = \sum_{i=1}^{n} w_i \mathbf{a}_{t,i}, \tag{5}$$

where $\mathbf{a}_{t,i}$ is the action predicted by the i-th expert. This action corresponds to the desired joint positions of the robot.

The desired joint positions are then converted into joint torques using a PD controller:

$$\boldsymbol{\tau} = K_p(\hat{\boldsymbol{q}} - \boldsymbol{q}) + K_d(\hat{\dot{\boldsymbol{q}}} - \dot{\boldsymbol{q}}), \tag{6}$$

where $\hat{\boldsymbol{q}}$ and $\hat{\dot{\boldsymbol{q}}}$ are the desired joint positions and velocities, $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ are the current joint positions and velocities, and K_p and K_d are the proportional and derivative gains.

4) Auxiliary Loss: In preliminary experiments, we observed that the gating network tends to collapse to a single expert without additional regularization, leading to poor specialization and reduced recovery performance. To mitigate this, we introduce an auxiliary loss consisting of two complementary components: a load-balancing term and a diversity-promoting term.

- **Load-balancing Loss.** To prevent expert underutilization, we encourage the average expert weights $\bar{p} \in \mathbb{R}^n$ to match a uniform distribution:

$$\mathcal{L}_{\text{balance}} = \frac{1}{n} \sum_{i=1}^{n} \left(\bar{p}_i - \tfrac{1}{n}\right)^2. \tag{7}$$

- **Diversity-promoting Loss.** While load balancing encourages fairness, it does not ensure that experts learn distinct behaviors. To promote specialization, we penalize low pairwise Jensen–Shannon divergence

(JSD) [26], [27] between expert assignment distributions:

$$\begin{aligned} \mathrm{JSD}(p \,\|\, q) &= \tfrac{1}{2}\,\mathrm{KL}(p \,\|\, m) + \tfrac{1}{2}\,\mathrm{KL}(q \,\|\, m)\,, \\ m &= \tfrac{1}{2}(p+q), \end{aligned} \tag{8}$$

where p and q are gating distributions and KL denotes the Kullback–Leibler divergence. Averaging across all pairs yields the diversity term $\mathcal{L}_{\text{div}} = \mathrm{JSD}(p \,\|\, q)$.

The auxiliary loss combines both terms:

$$\mathcal{L}_{\text{aux}} = \lambda_{\text{balance}}\mathcal{L}_{\text{balance}} + \lambda_{\text{div}}\mathcal{L}_{\text{div}}, \tag{9}$$

where λ_{balance} and λ_{div} control the relative contributions.

By enforcing balanced utilization and diverse specialization, this auxiliary loss stabilizes MoE training and improves the robustness of policy in various fall recovery scenarios.

B. Reward Design

Our reward function guides the robot to recover effectively and stably after a fall. The process is divided into three height-based stages: **Stage I**—body in ground contact exploring rising motions; **Stage II**—standing up; and **Stage III**—posture stabilization. Each stage has its own reward preference, and the total reward at time t is:

$$r(t) = w_p r_p(t) + w_m r_m(t) + w_s r_s(t)\phi(t) + w_c r_c(t), \tag{10}$$

where w_p, w_r, w_s, w_c are the respective weights for each component , $r_p(t)$, $r_m(t)$, $r_s(t)$, and $r_c(t)$ are the individual rewards defined in the following list, and $\phi(t)$ is the temporally optimized reward term for the stability stand-up reward. All reward functions is summarized in Table I.

- **Posture and height reward $r_p(t)$. (Stage I, II, III)** Encourages an upright posture by penalizing base orientation deviations and rewarding base height approaching the target.
- **Recovery motion regularization reward $r_m(t)$. (Stage I, II)** This component regularizes excessive limb excursions and enforces smoother joint coordination to prevent unnatural transitions, promoting physically plausible and efficient motion during the rising process.
- **Stability stand-up reward $r_s(t)$. (Stage III)** This component reinforces balance and robustness once the robot successfully maintains standing posture.
  We introduce a **temporally optimized reward** $\phi(t)$ to regulate standing duration:

$$\phi(t) = \begin{cases} \dfrac{t}{t_{\text{target}}} - 1, & 0 < t < t_{\text{target}}, \\ 1, & t \geq t_{\text{target}} \text{ or } t \leq 0, \end{cases} \tag{11}$$

  where t is the elapsed time since recovery begins and $t_{\text{target}} = 2\,\text{s}$. This term scales stability rewards to discourage overly fast rising; recoveries completed before t_{target} yield proportionally reduced rewards, promoting smoother and sustained transitions. Once $t \geq t_{\text{target}}$, the factor saturates at 1 with no further effect.
- **Regularization reward $r_c(t)$. (Stage I, II, III)** This component is introduced to ensure safe and smooth control, including penalties for excessive joint velocities, large joint torques, violations of joint position limits, and action smoothness.

TABLE I: Reward function elements. $\exp(\cdot)$ is exponential operators. The f_{tol} is a gaussian-style function with a saturation bound [28]. $\boldsymbol{h}$, $\boldsymbol{p}$, $\boldsymbol{g}$, $\boldsymbol{q}$ and $\boldsymbol{\tau}$ are the height of the body, body position, gravity vector projected into the robot's body frame, joint position and joint torque, respectively. In stability stand-up reward, σ is adaptive and thus varies according to the tracking performance, the values listed in the table correspond to the initial settings.

Reward	Expression	Weight
Posture and height	r_p	0.55
Head height	$f_{\text{tol}}\left(\boldsymbol{h}_{\text{head}}, [0.46, \infty], 0.46, 0.1\right)$	1.0
Base orientation	$f_{\text{tol}}\left(-\boldsymbol{g}^z_{\text{base}}, [0.99, \infty], 1, 0.05\right)$	1.0
Motion regularization	r_m	0.2
Hip deviation	$\mathbb{1}\left(\max(\lvert\boldsymbol{q}_{\text{hip}}\rvert) > 1.2\right) \mid \mathbb{1}\left(\min(\lvert\boldsymbol{q}_{\text{hip}}\rvert) > 0.7\right)$	-5
Foot orientation	$f_{\text{tol}}\left(-\boldsymbol{g}^z_{\text{foot}}, [0.8, \infty], 1.0, 0.05\right)$	10
Foot displacement	$\exp(-2(\boldsymbol{p}^{xy}_{foot} - \boldsymbol{p}^{xy}_{base})^2)$	2.5
Knee deviation	$\mathbb{1}\left(\max(\lvert\boldsymbol{q}_{\text{knee}}\rvert) > 1.65\right) \mid \mathbb{1}\left(\min(\lvert\boldsymbol{q}_{\text{knee}}\rvert) < -0.06\right)$	-0.25
Ankle parallel	$\left(\sum \text{var}\left(\boldsymbol{q}^z_{\text{ankle}}\right)\right)/2 < 0.05$	10
Foot distance	$f_{\text{tol}}\left((\boldsymbol{p}^{left}_{foot} - \boldsymbol{p}^{right}_{foot})^2, [0, 0.4], 0.3, 0.05\right)$	-10
Stability stand-up	r_s*(adaptive tracking sigma)*	0.2
Base angular velocity	$\exp(-\boldsymbol{\omega}^2_{xy}/0.5)$	10
Base linear velocity	$\exp(-\boldsymbol{v}^2_z/0.2)$	10
Base height	$\exp(-(\boldsymbol{h}_{base} - \boldsymbol{h}^{target}_{base})^2/0.05)$	10
Body posture	$\exp(-(\boldsymbol{q} - \boldsymbol{q}^{target})^2/10)$	10
Body deviation	$\exp(-(\boldsymbol{q}_{left} - \boldsymbol{q}_{right})^2/0.5)$	5
Regularization	r_c	0.05
Joint accelerations	$\ddot{\boldsymbol{q}}^2$	$-2.5e^{-7}$
Action rate	$(\mathbf{a}_t - \mathbf{a}_{t-1})^2$	-0.01
Smoothness	$(\mathbf{a}_t - 2\mathbf{a}_{t-1} + \mathbf{a}_{t-2})^2$	-0.01
Torques	$\boldsymbol{\tau}^2$	$-2.5e^{-6}$
Joint power	$\lvert\dot{\boldsymbol{q}} \times \boldsymbol{\tau}\rvert$	$-2.5e^{-5}$
Joint velocity	$\dot{\boldsymbol{q}}^2$	$-1e^{-3}$
Joint position limits	$\sum_i \left[\left(\boldsymbol{q}_i - \boldsymbol{q}^{\text{lower}}_i\right).\text{clip}(-\text{inf}, 0) + \left(\boldsymbol{q}_i - \boldsymbol{q}^{\text{higher}}_i\right).\text{clip}(0, \text{inf})\right]$	-100
Joint velocity limits	$\sum_i \left[\left(\lvert\dot{\boldsymbol{q}}_i\rvert - \dot{\boldsymbol{q}}^{\text{limit}}_i\right).\text{clip}(0, \text{inf})\right]$	-1

C. Curriculum Design

1) Initialization Posture Curriculum: Training covers four postures: supine, prone, left, and right lateral. Each posture is sampled with probability inversely proportional to its average standing height $\bar{h}$,

$$p_i = \frac{\frac{1}{\bar{h}_i}}{\sum_{j=1}^{4} \frac{1}{\bar{h}_j}} \tag{12}$$

so that harder postures are emphasized and easier ones gradually down sampled. As recovery heights equalize, the

distribution converges to uniform, yielding a balanced policy across diverse fallen states.

2) Auxiliary Force Curriculum: An upward assistive force is applied in early training when the torso is near vertical, increasing successful stand-ups and alleviating reward sparsity. The force is gradually reduced as the robot reaches the target height more consistently and is eventually removed, leaving a policy that recovers without assistance.

3) Velocity Constraint Curriculum: Joint and base velocity limits are initially loose to encourage exploration and the acquisition of feasible recovery strategies. These limits are progressively tightened to safe values, guiding the policy toward stable, physically plausible motions without excessive speeds or oscillations.

D. Motion Smoothness

Motion smoothness during recovery is enhanced by incorporating the Locally Lipschitz Continuous Constraint (L2C2) [29], which enforces local smoothness of the policy and value networks through regularization on interpolated states. Given two consecutive states $\mathbf{s}_t$ and $\mathbf{s}_{t+1}$, an interpolated sample is generated as $\hat{\mathbf{s}} = \mathbf{s}_t + (\mathbf{s}_{t+1} - \mathbf{s}_t) \cdot u,\ u \sim \mathcal{U}(\cdot)$. The regularization term is defined as

$$\mathcal{L}_{\text{L2C2}} = \lambda_\pi \|\pi_\theta(\mathbf{s}_t) - \pi_\theta(\hat{\mathbf{s}})\|_2^2 + \lambda_V \|V_\theta(\mathbf{s}_t) - V_\theta(\hat{\mathbf{s}})\|_2^2, \quad (13)$$

where π_θ and V_θ denote the policy and value networks. The $\mathcal{L}_{\text{L2C2}}$ loss enforces local smoothness in the policy, reducing jitter and producing more stable standing-up motions.

E. Adaptive Tracking Optimization

Postural stability during the standing phase is improved by employing an exponential form of tracking reward from PBHC [10], where task-specific tracking terms are scaled by a tracking factor σ that adjusts the reward's sensitivity to deviations in joint and base trajectories. Following [10], σ is optimized to minimize the accumulated tracking error along the reference trajectory, formulated as a bilevel optimization problem in which the error sequence of the converged policy determines the optimal σ^*. Using this optimal factor, the policy emphasizes reducing significant tracking deviations while avoiding excessive penalties for minor errors, which enhances overall stability, smooths motion transitions, and enables more robust maintenance of the upright posture under disturbances.

F. Domain Randomization

We mitigate the reality gap in bipedal fall recovery using extensive domain randomization [30] over inertial, actuation, sensing, and environmental parameters. Inertial randomization covers base mass (±2.0 kg), link mass scaling (0.95–1.05×), and CoM offsets (±0.04 m). Actuation is varied via motor strength (0.9–1.1×) and PD gains (0.8–1.2×). Environmental factors include ground friction (μ=0.1–1.0) and restitution (0–1), while initial conditions are diversified through joint offsets (±0.2 rad), scaling (0.8–1.2×), base pose perturbations, and external pushes (≤0.5 m/s).

TABLE II: Training hyper-parameter values.

Parameter	Value	Parameter	Value
Optimizer	Adam	Learning rate	1×10^{-3}
GAE parameter	0.95	Discount factor	0.99
Clip range	0.2	Batch size	4096×24
Desired KL-divergence	0.01	Max gradient norm	1.0
Number of Experts	6	$\lambda_{balance}$	0.03
λ_{div}	0.003	λ_π	1.0
λ_V	0.1		

Latency is randomized to capture actuation and observation delays (0–40 ms), with observation delays modeled separately as they dominate hardware performance. Without delay randomization, policies exhibit oscillations, whereas including both delays yields robust and stable recovery.

V. EXPERIMENTS AND RESULTS

We evaluate the proposed MoE-based fall recovery policy through simulation and real-world experiments, examining recovery performance in diverse postures, the impact of each component through ablation studies, and the robustness and generalization of the policy under challenging conditions.

A. Implementation Details

1) Training and Simulation: The Moe policy is implemented in PyTorch [31] and trained with PPO [23] in IsaacGym [32], using 4,096 parallel agents and the hyperparameters in Table II. A PD controller runs at 200 Hz in simulation and 500 Hz on hardware, while policies execute at 50 Hz. Before deployment, the trained models are validated in Mujoco [33]. All experiments are conducted on a workstation with an NVIDIA RTX 4070 Ti Super GPU, 32 GB RAM, and an AMD Ryzen 7 9700X CPU.

2) Real-World Setup: Our 7 kg bipedal robot has 10 actuated DOF (5 per leg) with 0.1 m thigh and calf segments, yielding a 0.35 m nominal height. Each joint uses torque-controlled actuators (5 Nm continuous, 14 Nm peak). The system employs EtherCAT motor communication at 1 kHz, a 50 Hz real-time controller on an Intel NUC12WSHi7, and a 6-axis IMU for state estimation at 500 Hz.

B. Simulation Experiment

1) Metrics: We employ the following quantitative metrics to evaluate the performance of the learned standing-up policy:

- **Success Rate (S, %):** The proportion of episodes in which the robot successfully reaches an upright standing posture without subsequently falling down.
- **Average Feet Movement (M_f, cm^2):** The mean cumulative squared displacement of both feet in the horizontal plane after standing.
- **Average Smoothness (S_m, deg):** The sum of squared second-order differences of joint positions, normalized by the number of executed actions, capturing the smoothness of the generated motion.
- **Average Energy Consumption (E, J):** The sum of the absolute product between joint velocities and joint torques.

TABLE III: Simulation evaluation of trained policies. Baseline comparison includes privileged policies; ablation studies examine the effect of removing modules.

Methods	Success (%)	$M_f\downarrow$	$S_m\downarrow$	$E\downarrow$	$S_\tau\uparrow$	$S_\theta\uparrow$	$V_e\downarrow$
Baselines							
HoST [6]	73.49±0.62	1.6431±0.15	0.8716±0.04	52.5937±1.48	0.9623±0.002	0.8777±0.001	-
Ours with 6 experts	**99.62**±0.31	1.0206±0.09	**0.7536**±0.04	**43.9755**±1.26	**0.9703**±0.001	**0.9061**±0.002	**0.19**±0.03
Ablations							
Ours w/o MoE	80.17±0.49	1.7241±0.12	0.8650±0.05	46.6653±1.33	0.9598±0.002	0.8235±0.002	-
Ours with 2 experts	81.45±0.35	1.2511±0.10	0.8331±0.05	47.8314±1.42	0.9613±0.003	0.8419±0.003	0.20±0.03
Ours with 4 experts	92.17±0.49	1.0532±0.15	0.7950±0.04	45.5318±1.36	0.9658±0.002	0.8654±0.002	0.25±0.02
Ours with 8 experts	99.55±0.35	1.0135±0.12	0.7632±0.05	43.9823±1.31	0.9698±0.003	0.9042±0.001	0.23±0.02
Ours w/o height estimator	92.88±0.52	1.8558±0.14	0.8569±0.04	48.7522±1.45	0.9567±0.003	0.8148±0.001	0.28±0.02
Ours w/o CNN encoder	94.71±0.57	1.2847±0.15	0.8539±0.03	50.2156±1.51	0.9587±0.002	0.8328±0.003	0.27±0.03
Ours w/o auxiliary loss	95.46±0.60	1.8679±0.09	0.9305±0.05	57.5268±1.58	0.9567±0.002	0.8321±0.001	0.45±0.03
Ours w/o time-based shaping term	98.37±0.55	1.0146±0.11	0.8070±0.06	49.8399±1.40	0.9575±0.002	0.8217±0.001	0.23±0.03
Ours w/o adaptive tracking optimization	98.76±0.58	1.0459±0.14	0.8092±0.07	47.4311±1.48	0.9630±0.001	0.8359±0.003	0.23±0.02

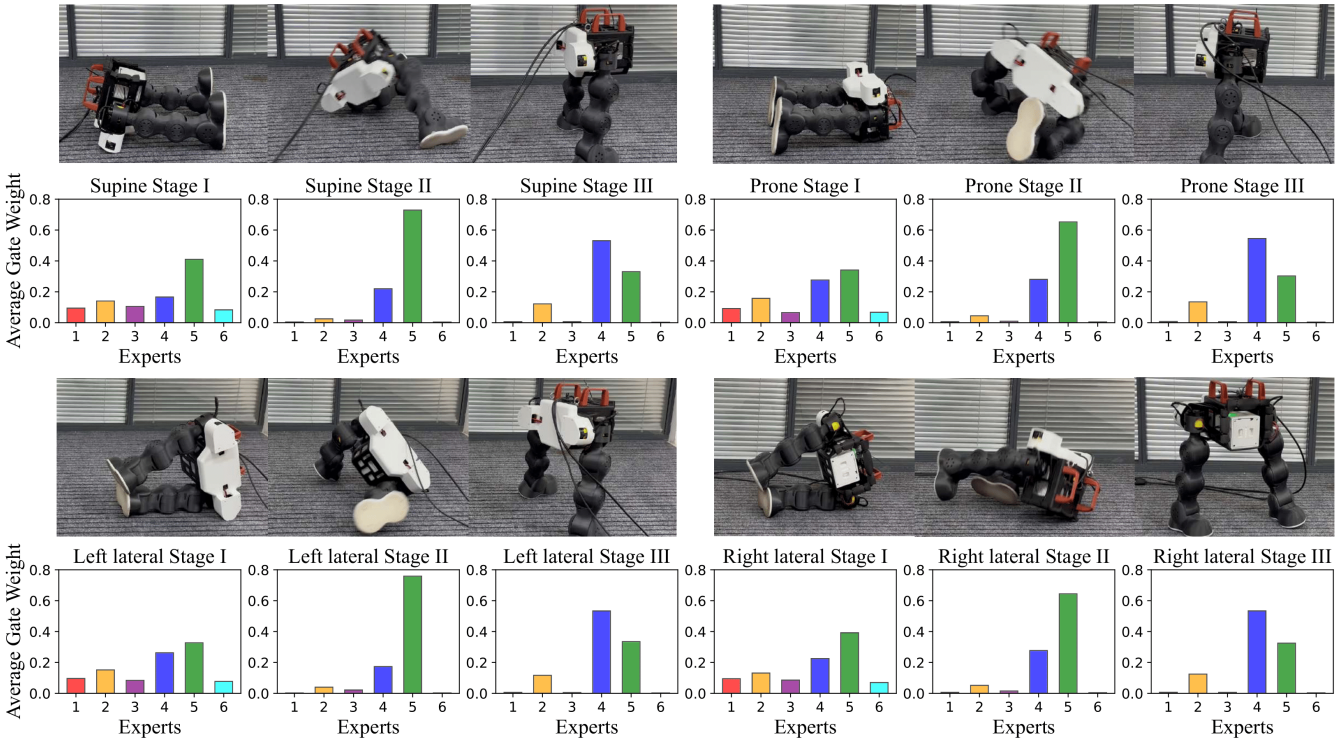

Fig. 3: **Expert usage in different scenarios.** Each subplot corresponds to the weights of multiple expert networks under a specific scenario. We divide the process into 12 scenarios based on body height and initial posture. Three stages and four initial postures are defined in IV.

- **Average Safety Scores (S_τ, S_θ):** Following [7], we adopt two safety scores that quantify the relative magnitude of commanded signals compared to their physical limits. Specifically, S_τ measures the normalized torque usage with respect to the torque limits, and S_θ measures the normalized DoF usage with respect to the joint position limits. For consistency and comparability, we adopt the same normalization ranges and limit parameter as defined in [7].
- **Variance of the Expert Weights (V_e):** The variance of the expert weights output by the gating network reflects the specificity and specialization of the expert selection.

2) Baselines and Ablations: To comprehensively evaluate the effectiveness of our proposed method, we compare it against baseline HoST [6] and conduct extensive ablation studies to understand the contribution of each key component.

- **HoST**: Robot rising up using HoST [6].
- **Ours with 6 experts**: Our policy trained with all modules.
- **Ours w/o MoE**: Our policy trained using a simple MLP as the actor network.
- **Ours with different expert numbers**: To study the effect of expert specialization, we train policies with 2, 4, and 8 experts.
- **Ours w/o height estimator**: The height estimator is removed, such that the gating network no longer has access to height information.

Fig. 4: Snapshots of the robot recovering in four distinct, randomized, and challenging scenarios. Rows (a) and (b) show trials initialized from randomized poses on the ground while rows (c) and (d) show trials initialized on the slope.

- **Ours w/o CNN encoder**: The CNN encoder which encodes the observations is removed.
- **Ours w/o auxiliary loss**: Auxiliary loss is not utilized in our MoE network.
- **Ours w/o time-based shaping term**: The time-based shaping term is removed from the target reward, reducing the constraint on the robot's rising speed.
- **Ours w/o adaptive tracking optimization**: The adaptive tracking factor is removed from the target reward.

3) Results and Analysis: The effectiveness of the proposed approach is evaluated through comparisons with several baselines and a series of ablation studies. All results are reported as averages over 4,000 simulated fall-recovery trials across diverse initial postures, as summarized in Table III.

Compared with the baseline, the complete method consistently achieves higher success rates and smoother recoveries across diverse conditions. Notably, HoST [6] succeeds in only three out of the four tested postures, whereas our approach achieves reliable recovery across all postures, highlighting the effectiveness of the overall design.

Replacing the MoE with a simple MLP leads to a clear drop in performance, indicating the importance of expert specialization for handling diverse states. Ablation on the number of experts shows that 2 lack specialization, 4 remain insufficient for diverse postures, and 8 yield marginal gains with higher computational cost and instability. We thus adopt 6 experts as a balanced configuration, providing sufficient specialization with minimal overhead. Removing the height estimator reduces stability during the standing phase, as the gating network no longer receives direct height information. The CNN encoder is found to be critical for capturing temporal dependencies in the observation stream and compressing them into a compact representation that fuses effectively with height information; without it, the policy fails to exploit temporal context and produces less stable recoveries. Eliminating the time-based shaping term results in less smooth motions and more jittery behaviors. The auxiliary loss ensures balanced training across all experts, and its removal reduces overall success rates. Finally, omitting adaptive tracking optimization increases tracking error and postural jitter, highlighting the role of the tracking factor in maintaining stability.

C. Real-World Experiment

We conduct real-world experiments on our bipedal robot to evaluate the effectiveness and robustness of the proposed framework. The experiments are organized into three parts:

1) Gating Network Insights: The robot is initialized in four distinct fallen postures (supine, prone, left-lateral, and right-lateral) and commanded to recover using the learned policy. As shown in Fig. 3, the gating network's weight distribution reveals a clear functional division: Expert 5 consistently dominates the lifting phase (Stage II), while Expert 4 governs the final stabilization phase (Stage III). The initial phase, by contrast, exhibits more distributed activations, reflecting the need to accommodate posture and contact-specific variations. These results demonstrate that the MoE architecture enables functional specialization across experts and dynamic allocation by the gating network, yielding reusable, phase-specific sub-policies. This mechanism allows a single learned policy to achieve robust recovery from diverse fall configurations, highlighting the effectiveness of MoE in multi-posture fall recovery.

2) Robust Recovery across Diverse Postures: As shown in Fig. 1, after standing up the robot is deliberately pushed into diverse fallen configurations, including challenging postures. A single unified policy consistently enables stable recovery

without retraining, demonstrating robustness to repeated and varied perturbations.

3) Robustness under Challenging Conditions: The robot is evaluated in highly demanding scenarios, including twisted or sprawled leg configurations that were never encountered during training, as well as recovery on a 15° inclined slope. As shown in Fig. 4, the policy consistently enables smooth and stable standing across all these cases, highlighting its strong robustness and ability to generalize to both unseen postures and difficult terrains.

VI. CONCLUSIONS

This paper presents a unified framework for multi-posture fall recovery in bipedal robots. It combines a MoE policy with a height estimator and historical proprioceptive information to dynamically allocate recovery tasks to specialized experts. Temporal reward optimization, a joint-velocity curriculum, and adaptive tracking enable smooth, stable, and natural stand-up motions. The framework can be easily integrated into existing robotic deployment pipelines, allowing the robot to recover from falls during task execution—such as walking—without human intervention. Extensive simulation and real-world experiments demonstrate robust recovery across diverse fall postures, and the policy can be deployed zero-shot on hardware, highlighting both its generalization and practical feasibility.

REFERENCES

[1] R. Sutton and A. Barto, "Reinforcement learning: An introduction," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 1054–1054, 1998.

[2] S. Ha, J. Lee, M. van de Panne, Z. Xie, W. Yu, and M. Khadiv, "Learning-based legged locomotion: State of the art and future perspectives," *The International Journal of Robotics Research*, vol. 44, no. 8, pp. 1396–1427, 2025.

[3] T. He, J. Gao, W. Xiao, *et al.*, "Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills," *Robotics: Science and Systems (RSS)*, June 2025.

[4] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. M. Kitani, C. Liu, and G. Shi, "Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning," *Conference on Robot Learning*, 2024.

[5] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control," *The International Journal of Robotics Research*, vol. 44, no. 5, pp. 840–888, 2025.

[6] 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," *Robotics: Science and Systems (RSS)*, June 2025.

[7] X. He, R. Dong, Z. Chen, and S. Gupta, "Learning getting-up policies for real-world humanoid robots," *Robotics: Science and Systems (RSS)*, June 2025.

[8] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.

[9] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *International Conference on Learning Representations*, 2017.

[10] W. Xie, J. Han, J. Zheng, H. Li, X. Liu, J. Shi, W. Zhang, C. Bai, and X. Li, "Kungfubot: Physics-based humanoid whole-body control for learning highly-dynamic skills," *arXiv preprint arXiv:2506.12851*, 2025.

[11] R. Huang, S. Zhu, Y. Du, and H. Zhao, "Moe-loco: Mixture of experts for multitask locomotion," *arXiv preprint arXiv:2503.08564*, 2025.

[12] D. Wang, X. Wang, X. Liu, J. Shi, Y. Zhao, C. Bai, and X. Li, "More: Mixture of residual experts for humanoid lifelike gaits learning on complex terrains," *arXiv preprint arXiv:2506.08840*, 2025.

[13] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: Adversarial motion priors for stylized physics-based character control," *ACM Transactions on Graphics (ToG)*, vol. 40, no. 4, pp. 1–20, 2021.

[14] Y. Li, Y. Lin, J. Cui, T. Liu, W. Liang, Y. Zhu, and S. Huang, "Clone: Closed-loop whole-body humanoid teleoperation for long-horizon tasks," *Conference on Robot Learning*, 2025.

[15] M. Khorram and S. A. A. Moosavian, "Balance recovery of a quadruped robot," in *2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM)*. IEEE, 2015, pp. 259–264.

[16] J. Stückler, J. Schwenk, and S. Behnke, "Getting back on two feet: Reliable standing-up routines for a humanoid robot," in *Proceedings of the Ninth International Conference on Intelligent Autonomous Systems (IAS 2006)*. IOS Press, 2006, pp. 676–685.

[17] K. Araki, T. Miwa, H. Shigemune, S. Hashimoto, and H. Sawada, "Standing-up control of a fallen humanoid robot based on the ground-contacting state of the body," in *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2018, pp. 3292–3297.

[18] Y. Lu, Y. Dong, J. Zhang, J. Ma, and P. Lu, "Fr-net: Learning robust quadrupedal fall recovery on challenging terrains through mass-contact prediction," *IEEE Robotics and Automation Letters*, vol. 10, no. 7, pp. 6632–6639, 2025.

[19] B. Deng, L. Rossini, J. Wang, W. Wang, and N. Tsagarakis, "Learning to recover: Dynamic reward shaping with wheel-leg coordination for fallen robots," *arXiv preprint arXiv:2506.05516*, 2025.

[20] 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.

[21] P. Chen, Y. Wang, C. Luo, W. Cai, and M. Zhao, "Hifar: Multi-stage curriculum learning for high-dynamics humanoid fall recovery," *arXiv preprint arXiv:2502.20061*, 2025.

[22] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," in *Robotics: Science and Systems*, June 2018.

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

[24] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[25] Y. Kim, "Convolutional neural networks for sentence classification," *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, oct 2014.

[26] M. L. Menéndez, J. A. Pardo, L. Pardo, and M. d. C. Pardo, "The jensen-shannon divergence," *Journal of the Franklin Institute*, vol. 334, no. 2, pp. 307–318, 1997.

[27] B. Fuglede and F. Topsoe, "Jensen-shannon divergence and hilbert space embedding," in *International Symposium onInformation Theory (ISIT)*, 2004, pp. 31–.

[28] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. d. L. Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, *et al.*, "Deepmind control suite," *arXiv preprint arXiv:1801.00690*, 2018.

[29] T. Kobayashi, "L2c2: Locally lipschitz continuous constraint towards stable and smooth reinforcement learning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4032–4039.

[30] J. Tobin *et al.*, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.

[31] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: an imperative style, high-performance deep learning library," in *Neural Information Processing Systems*, 2019.

[32] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[33] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.