

# Unified Humanoid Fall-Safety Policy from a Few Demonstrations

Zhengjie Xu, Ye Li, Kwan-Yee Lin, Stella X. Yu

University of Michigan

{zhengjie, yeyli, junyilin, stellayu}@umich.edu

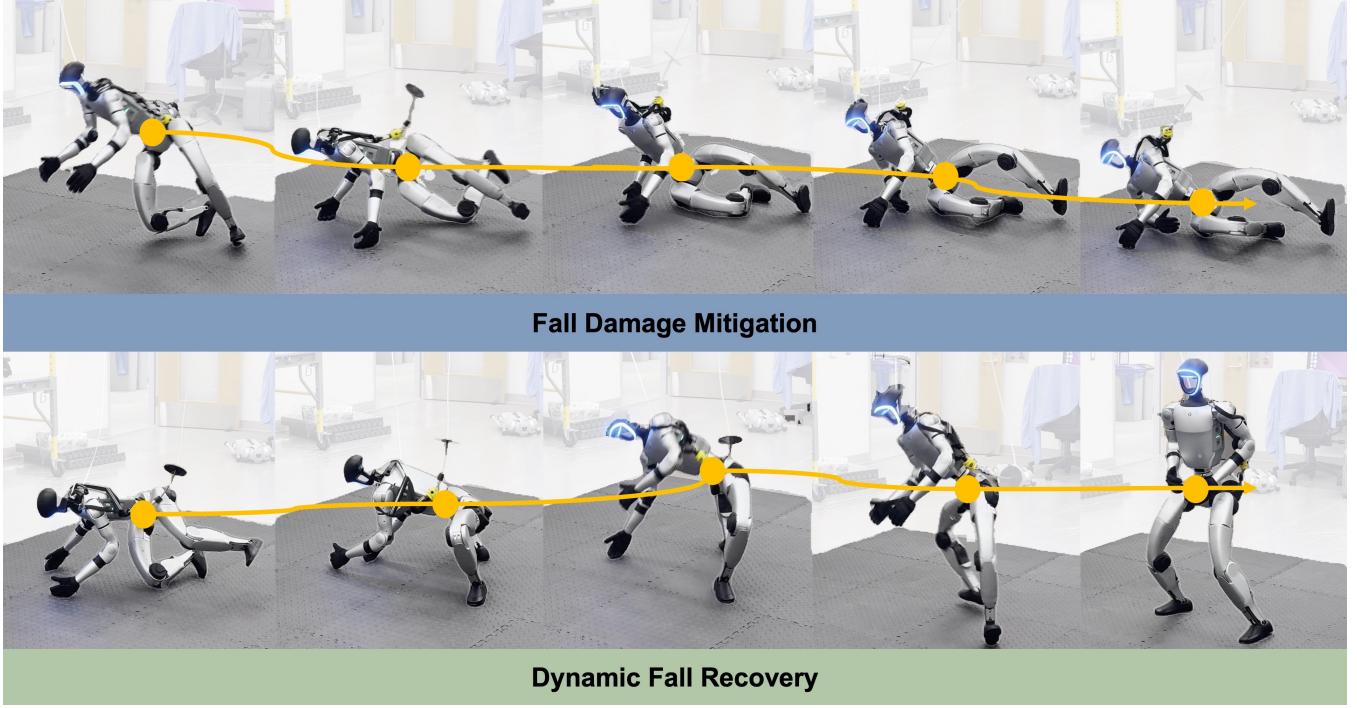


Fig. 1. Our method enables humanoids to fall safely and rise promptly. Snapshots show real-world deployment on the Unitree G1: When suddenly destabilized, the robot redirects into a side fall with arm buffering, then reorients and rises, demonstrating adaptive and resilient recovery.

**Abstract**— Falling is an inherent risk of humanoid mobility. Maintaining stability is thus a primary safety focus in robot control and learning, yet no existing approach fully averts loss of balance. When instability does occur, prior work addresses only isolated aspects of falling: avoiding falls, choreographing a controlled descent, or standing up afterward. Consequently, humanoid robots lack integrated strategies for impact mitigation and prompt recovery when real falls defy these scripts. We aim to go beyond keeping balance to make the entire fall-and-recovery process safe and autonomous: prevent falls when possible, reduce impact when unavoidable, and stand up when fallen. By fusing sparse human demonstrations with reinforcement learning and an adaptive diffusion-based memory of safe reactions, we learn adaptive whole-body behaviors that unify fall prevention, impact mitigation, and rapid recovery in one policy. Experiments in simulation and on a Unitree G1 demonstrate robust sim-to-real transfer, lower impact forces, and consistently fast recovery across diverse disturbances, pointing toward safer, more resilient humanoids in real environments. Videos are available at <https://firm2025.github.io/>.

## I. INTRODUCTION

Where there are legs, there will be stumbles. Even the most carefully trained humanoids - built for agile locomotion and intelligent navigation planning - are bound to be jolted off balance by a stray push, a loose stone, or an unexpected gust.

When a 1.3 m, 35 kg Unitree G1 robot with delicate vision and force sensors topples, the damage can be costly: bent joints, cracked housings, and extended downtime.

Such incidents are not rare anomalies but fundamental risks of legged mobility. Balance controllers can reduce but never eliminate unexpected falls [1], [2]. Unlike wheeled or quadruped robots, which enjoy wider and more stable support base [3], [4], humanoids combine tall, narrow frames with dozens of degrees of freedom, producing diverse and hard-to-predict fall dynamics [5], [6].

We aim to give humanoids a single instinct for self-preservation: a unified policy that keeps them upright whenever possible and, when a fall is unavoidable, ensures they fall safely and rise on their own (Fig. 1).

Prior work tackles only isolated pieces of this chain. Classical balance controllers focus on avoiding falls altogether [1], [2], motion-planning methods choreograph a controlled descent [6], [7], and recovery studies begin only after the damage is done, teaching robots to stand up from static supine postures [8], [9].

**Yet falling and rising are inseparable phases of a single physical process:** How a robot falls directly shapes how it

can get back up. By unifying mitigation and recovery, our approach explicitly addresses this coupled dynamic.

The challenge is daunting. Once balance is lost, a fall becomes a complex, high-dimensional physical process with rapidly changing contacts and forces, exposing weaknesses in both major camps of humanoid control:

- 1) **Model-based control.** Carefully planned motions can be computed for particular impacts [6], [7], but such methods depend on simplified dynamics and become intractable as the range of disturbances grows.
- 2) **Learning-based control.** Imitation learning typically requires dense, full-motion demonstrations, which are difficult to collect at scale and often lead to policies that collapse to fixed reference trajectories with poor adaptability [10]. Reinforcement learning (RL) must juggle a set of carefully crafted reward terms whose interactions are hard to anticipate, making reward engineering difficult and often producing brittle or unnatural behaviors [11], [12]. Without an effective way to represent multi-modal policies (e.g., through skill embeddings or generative models), RL struggles to encode the diverse actions needed for safe falling and rising [13], [14].

Due to these limitations, no prior method reliably spans the full spectrum from balance maintenance, through damage-mitigating fall, to autonomous recovery.

We tackle this challenge with learning a single, unified humanoid fall-safety policy from just a few demonstrations. By fusing sparse human demonstrations with reinforcement learning and an adaptive diffusion-based memory of safe reactions, we learn adaptive whole-body behaviors that cover fall prevention, impact mitigation, and rapid recovery within a single policy (Fig. 2). The policy learning proceeds in two stages: *learning safe skill priors* and *learning adaptiveness*, achieved through the following four steps:

- 1) **Seed safe skill acquisition.** The robot begins with a few temporally sparse human key poses, internalizing them through RL to fit its own morphology and dynamics. This creates dense reaction trajectories that seed safe falling and rising in its action space.
- 2) **Safe skill enrichment.** Targeted stitching of compatible falling and rising motions, combined with policy rollouts, generates additional safe trajectories. This expansion yields strategies for pre-emptive fall prevention, diverse fall variations, fall mitigation, and reliable recovery.
- 3) **Safe reactive memory.** All safe reactions are distilled into a diffusion policy that captures a rich, multi-modal distribution of fall-and-rise behaviors. A learned feature predicts the next safe target pose from past trajectory data.
- 4) **Adaptive safe control.** At run time, the feature is extracted online to retrieve the nearest neighbour from a memory bank of safe poses. Refreshing predictions at every step, the system assembles safe trajectories on the fly from overlapping segments, expanding each target into a neighborhood of possibilities and enabling rapid adaptation to unforeseen terrain or disturbances.

The result is a humanoid that does more than stay on its feet. It anticipates trouble, redirects unavoidable falls to minimize

harm, and rebounds swiftly to a stable stance, turning an inevitable weakness into evidence of genuine resilience.

Experiments in simulation and on the Unitree G1 confirm robust sim-to-real transfer, with lower impact forces and prompt, reliable recovery across diverse disturbances [8], [9], [11]. By unifying pre-emptive fall prevention, impact mitigation, and rapid recovery within a single memory-driven policy, our approach advances safe humanoid control and lays a strong foundation for resilient service and assistive robots in unstructured environments.

## II. RELATED WORK

### A. Humanoid Control

Model-based methods laid the foundations of humanoid control [3], [15]. Learning-based methods have since advanced the field, from IL [16] to RL [11], [12]. Human demonstrations further enrich motion style and diversity [13], [17]–[19]. Recent studies extend locomotion to challenging motions and diverse terrains [20]–[25]. Our work complements these advances with a unified learning framework that goes beyond locomotion to prevent falls, mitigate impact, and recover robustly without heavy reward engineering.

### B. Humanoid Fall Mitigation

Early methods imitate human break-falls to limit damage [5], [26]–[28], but rely on heuristics and offline tuning. Model-based methods cast safe falling as momentum redirection, trajectory optimization, or multi-contact planning [7], [29]–[31], and energy-based controllers provide online shaping [32], [33]. Mechanical or control compliance lengthens impact and regulates post-impact behavior [34], [35], while direction-control strategies steer the body toward safer contact regions [36], [37]. These methods work in targeted scenarios but typically depend on hand-crafted strategies, simplified dynamics, or pre-specified contact sequences; in contrast, our work learns a unified policy that generalizes across disturbances, enabling pre-emptive fall prevention, impact mitigation, and prompt recovery within one framework.

### C. Humanoid Fall Recovery

Classical model-based approaches plan stand-up motions after a fall [5], [27], [38], [39], but they generalize poorly and are sensitive to disturbances. Learning-based methods improve robustness: Some imitate predefined trajectories [13], [17], [40], while others train policies from scratch [8], [9], [41]. Although these works broaden the range of recoverable postures, resulting motions often remain unnatural and fragile. A recent quadruped study jointly addressed falling and recovery [4], but no prior humanoid work unifies fall mitigation and prompt recovery. Our work closes this gap by integrating all into a single policy.

### D. Diffusion Models in Robotics

Diffusion models have recently been adopted for control and planning by casting policy learning as conditional generative modeling [14], [42], [43]. Building on these foundations, legged-robot studies learn multi-skill policies from

offline data and deploy them online. For example, Diffuse-Loco achieves robust zero-shot transfer for quadruped locomotion [44], while preference alignment and test-time guidance improve robustness in out-of-distribution states [45]. Hybrid approaches embed MPC for constraint satisfaction and safety [46], [47]. Our work leverages diffusion to encode a multi-modal memory of safe fall-and-rise behaviors.

### III. PROBLEM FORMULATION

We study the problem of *fall damage mitigation and recovery* for humanoid robots in unstructured environments, formulated as a *dynamic process* that begins with a destabilizing disturbance to fall and ends once the robot regains a stable upright pose above a target height. This process inherently involves two levels: 1) *damage mitigation* (minimizing impact during the fall), and 2) *recovery* (standing back up)—but unlike prior works, we do not separate them; instead, we learn the coupled process directly. At each timestep  $t$ , we perceive the robot proprioception information to feed into the policy to output action  $a_t \in \mathbb{R}^{23}$ , which are offset applied to the robot’s nominal joint configuration  $q^{\text{default}}$ . By learning this unified dynamic process, a single policy can be directly applied to three tasks: *fall mitigation only*, *recovery only*, and the *full coupled problem*, without task-specific retraining.

### IV. METHOD

We present *FIRM*, (short for *fall mitigation and recovery* from a few human *demonstrations*), a control policy for the diverse, complex dynamics of humanoid falling and recovery (Fig. 2). FIRM unifies fall mitigation and recovery in a single framework that balances *safety* and *behavioral adaptiveness*. It operates in two stages. **1) Skill priors** (Sec. IV-A): a few human demonstrations are fitted and retargeted to the G1 humanoid, then sparsified into key frames (Fig. 2a). These seed skills are expanded with RL-based augmentation and post-stitching (Fig. 2b) to produce diverse, damage-reducing trajectories. **2) Adaptive memory** (Sec. IV-B): the enriched trajectories are distilled into a diffusion model and paired with a lightweight adapter (Fig. 2c) that enables real-time fall mitigation and recovery across varied conditions.

#### A. Learning Fall-and-Recover Skill Priors

**1) Collecting Seed safe skill via Retargeting Human Videos:** We expect the task controller to prioritize safety during falling and recovery, which requires safe motion patterns. Since such patterns are difficult to engineer manually, we utilize human demonstrations, which naturally encode safety-critical behaviors. However, current large-scale MoCap datasets like AMASS [48] lack realistic fall–recovery motions. To address this gap, we collect a small number of monocular human video demonstrations, and process them through fitting to SMPL [49] and retargeting to the G1 humanoid. In total, we use 5(2 : 2 : 1) high-quality trajectories covering forward, sideways, and backward fall–recover processes on flat ground, collected from subjects of different genders and heights to provide variety in motion styles.

While the data volume of demonstrations is small in scale, its quality and targeted varieties are crucial, as they provide sufficient prototypes for our later designs to compose and expand upon. Each trajectory includes the information of all joint positions  $q_i$ , rigid body poses relative to root  $T_{b_i}$  and root poses  $T_{\text{root}}$  retargeted to G1 robot. To obtain velocities, we further calculate joint velocities  $\dot{q}_i$ , rigid body twists relative to root  $V_{b_i}$ , and root twist  $V_{\text{root}}$  using finite difference.

#### 2) Expanding Priors via Sparse-key-frame Augmentation

**Policy Learning:** Unlike conventional motion-tracking tasks, directly fitting recorded trajectories is insufficient for motions involving rich contacts and lacks adaptability to different environments. Furthermore, strategies for fall damage reduction and recovery may differ between humans and robots due to differences in morphology and actuation. To address these issues, we formulate the prior learning problem as a *sparse key-frame tracking task in a goal-conditioned RL learning form*, which can provide safety-critical posture anchors while leaving flexibility for the RL policy to explore and optimize its behaviors according to the robot’s own dynamics.

The goal of this stage is thus to track a sequence of sparse key-frames, reach a final standing configuration, and minimize fall damage throughout the process. We formulate it as a finite-horizon control problem with horizon  $H = 10$ s, deliberately chosen to exceed the length of any collected demonstration, so that the robot must not only follow the motion but also maintain a stable standing posture afterwards. Formally, we define a sparse trajectory as  $\mathcal{P} = \{P_1, P_2, \dots, P_N, P_{\text{stand}}\}$ , where each frame  $P_n$  is represented by  $P_n = \{q_{i,n}, \dot{q}_{i,n}, T_{b_i,n}, T_{\text{root},n}, V_{b_i,n}, V_{\text{root},n}\}$ , with joint states  $(q_{i,n}, \dot{q}_{i,n})$ , root and link poses  $(T_{\text{root},n}, T_{b_i,n})$ , and corresponding twists  $(V_{\text{root},n}, V_{b_i,n})$ , sampled at a fixed frequency  $f$ . The control objective requires the robot to reach each successive key-frame  $P_{n+1}$  within the interval  $[t_n, t_{n+1}]$ , and finally hold the standing frame  $P_{\text{stand}}$  until the episode terminates. For each trajectory, we train a corresponding augmentation policy. The training details are outlined as follows.

**State Initialization and Rollout.** At the beginning of each episode, we randomly pick a frame  $P_0$  from the *dense* trajectory, and copy all the joint and root information to *initialize* the robot’s state, while using only sparse key-frames for motion tracking. To simulate diverse fall conditions, we then randomize the base and joint states slightly and disable the actuators for a random duration  $t \sim \mathcal{U}(0.04, 1.0)$ s, allowing the robot to enter free fall before regaining control.

As the control and video frequency are not the same, we linearly interpolate all the information for the time between two frames. The final standing pose is set to be G1 robot’s default pose with root height at 0.8m. The height is slightly higher than the robot’s actual standing height of  $\sim 0.728$ m. This margin encourages the policy to stand more upright and avoid slouched postures during recovery. We keep the root yaw of the last frame the same as the trajectory’s end, avoiding unnecessary rotation to a world-neutral orientation. Since retargeted trajectories may drift in root position, occasionally placing the robot above or below the ground, we preprocess each frame using forward kinematics and shift

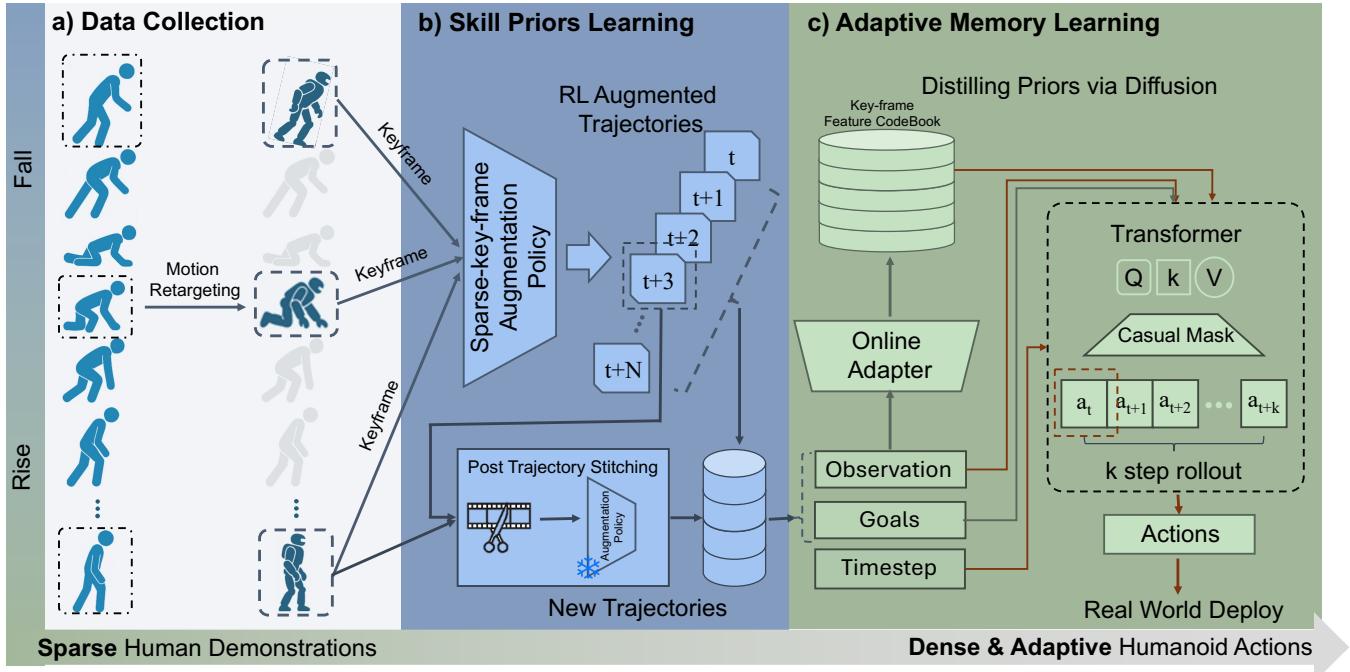


Fig. 2. **Workflow Overview.** From a few sparse human key poses, the robot seeds safe fall–rise skills, expands them through RL with post-trajectory stitching, distills enriched behaviors into a diffusion-based action memory, and composes online adapter to execute actions with context-awareness.

the root height by the lowest Z-coordinate among all rigid bodies, followed by a  $0.05m$  offset to ensure clearance. This adjustment does not introduce harmful discontinuities, as the policy only follows sparse key-frames rather than dense trajectories, and in fact improves exploration between frames. **Policy Optimization.** We optimize our policy using an asymmetric actor-critic framework with PPO [13], [17]. In this design, the critic has access to privileged information from the simulator that is unavailable to the actor and real-world setting. The actor’s observation space consists of: 1) root angular velocity  $\omega_{root}$ , 2) joint position and velocities  $q, \dot{q}$ , 3) last actions  $a_{t-1}$ , 4) joint position difference with respect to next key frame  $q - q_{key}$  and 5) phase  $\phi$ . The critic network additionally observes root linear velocity  $v_{root}$ . The phase  $\phi$  is calculated by dividing the current runtime  $t$  by the total length of the trajectory in time  $T$  and clipped to 1 for any time steps exceeding the trajectory length:  $\phi = \min(t/T, 1)$ .

**Domain Randomization.** To further enhance the robustness of our policy and for later sim-to-real deployment, we followed previous works [8] to adopt domain randomization during our policy training. We randomize the friction ( $\mathcal{U}(0.25, 1.75)$ ), payload ( $\mathcal{U}(-1, 1)$ ), and gains for each joint (p-gain:  $\mathcal{U}(0.9, 1.1)$ , D-gain:  $\mathcal{U}(0.9, 1.1)$ ), and randomly push the robots. We trained our robots on rough terrains to make our policy more robust in various environments. Observation noise is also added in simulation.

**Episode Termination.** As this task is contact-rich, we do not terminate episodes upon collisions, except when joint or root velocity limits are exceeded. In many locomotion tasks, episodes are typically terminated when the base height drops below a threshold or when collisions occur, to ensure the robot remains in valid poses. These criteria are unsuitable

here, since our setting explicitly requires the robot to fall to the ground and recover. Likewise, motion-tracking tasks [22] often use termination based on deviations from reference trajectories; however, because we only track very *sparse* key-frames and aim to encourage exploration between them, this signal is also inappropriate. Therefore, our episode termination is kept minimal, while safety and stability are instead encouraged through the reward design described below.

**Rewards.** We formulate our rewards design mainly in 3 categories: 1) *Tracking rewards*: These are main task rewards that track the difference between the current robot states and key-frame robot states, including joint positions and velocities, and rigid body poses and twists. Different from the trajectory information, where rigid body poses and twists are in local frame with respect to the robot root, here we calculate these quantities in world frame, which seamlessly integrates the tracking of root poses and twists as well. We calculate the reward using the function  $h(d; \sigma) = \exp(-d^2/\sigma)$ . 2) *Style rewards*: To penalize harmful and un-natural behaviors, we add this set of rewards to constrain on action rate, joint acceleration, torque values and out-of-limit joint behaviors. 3) *Fall damage reduction rewards*: In order to mitigate the damage when falling on the ground, we add penalizing rewards on body collision, momentum change, and body yank as described in [4]. The scale and exact definition of the rewards can be found in Table I. In contrast to conventional fall recovery methods, our policy can utilize the safe motion pattern priors from human demonstrations to constrain fall recovery learning without complicated reward designs.

3) *Post trajectory stitching scheme*: In real-world scenarios, losing balance does not always lead to a complete fall, as humans often adjust themselves and quickly regain stability. However, our human demonstrations only cover trajectories

TABLE I

REWARD TERMS SUMMARY FOR PRIOR LEARNING. **TRACKING / STYLE**  
**/ FALL-DAMAGE REDUCTION** REWARDS.

Reward Term	Definition	Scale
Rigid body position tracking	$h(\sum_B w_B (T_{B,w} - \hat{T}_{B,w})^2; \sigma)$	1.25
Rigid body rotation tracking	$h(\sum_B (R_{B,w} - \hat{R}_{B,w})^2; \sigma)$	0.5
Rigid body linear velocity tracking	$h(\sum_B (v_{B,w} - \hat{v}_{B,w})^2; \sigma)$	0.125
Rigid body angular velocity tracking	$h(\sum_B (\omega_{B,w} - \hat{\omega}_{B,w})^2; \sigma)$	0.125
Joint position tracking	$h(\sum_j (q_j - \hat{q}_j)^2; \sigma)$	0.5
Joint velocity tracking	$h(\sum_j (\dot{q}_j - \hat{\dot{q}}_j)^2; \sigma)$	0.125
Joint position limit	$\sum_j \max(0,  q_j - q_j^{\text{limit}} )$	-10
Joint velocity limit	$\sum_j \max(0,  \dot{q}_j - \dot{q}_j^{\text{limit}} )$	-5
Action rate	$\sum (a[t] - a[t-1])^2$	$-1e^{-3}$
Torques	$\sum_j \tau_j^2$	$-1e^{-6}$
Acceleration	$\sum_j \dot{q}_j^2$	$-2.5e^{-7}$
Body collision	$\sum_B \ \lambda_B\ ^2$	$-1e^{-7}$
Momentum change	$\sum_B \ m_B a_B\ $	$-5e^{-3}$
Body yank	$\sum_B \ F_B\ ^2$	$-2e^{-6}$

where a fall actually occurs. Training and expanding solely on such data would cause the robot to treat any minor imbalance as a full fall, leading to overly conservative while risky behavior. To address this, we propose to reuse demonstration trajectories with shortcuts, to generate alternative balance-preserving rollouts. The assumption behind this is that the robots can regain balance under a range of perturbations as long as a suitable reference key-frame can be found and to be used as the anchor, *i.e.*, it corresponds to a feasible intermediate state observed in recovery phases. Therefore, instead of explicitly training the policy on every such near-balance trajectory, we construct them via stitching and allow the robot to follow these recomposed trajectories at test time. Concretely, for a randomly selected  $t < t_0$  (with  $t_0$  set to around one third of the trajectory length), we create a shortcut to a later key-frame  $t'$ , selected from the second half of the trajectory, whose root height  $h_{t'}$  is closest to the root height  $h_t$  and satisfies a clearance threshold of 0.05 m. The policy is then re-executed from  $s_t$  toward this new goal at  $t'$ , producing a stitched, new trajectory:  $G^{\text{new}} = \{(s_0, a_0), \dots, (s_t, a_t), (s_{t'}, a_{t'}), \dots, \}$ . This procedure allows trajectories to be recomposed beyond their original temporal order, encouraging the policy to connect more arbitrary trajectory states with later recovery strategies.

### B. Adaptive Memory Learning

1) *Distilling Priors via Diffusion Model:* We use the expert policies and the post trajectory stitching scheme to collect 4.5 million trajectory data pairs in the form of  $(o, g, a)$ , where  $o$  denotes observations,  $g$  represents reference sparse key-frames as goals, and  $a$  are the corresponding actions. We expect to distill these diverse trajectories into a single policy. Since the distribution of these pairs is inherently multi-modal, directly fitting a unimodal policy would collapse diverse strategies into averaged behaviors, leading to unnatural motions and degraded safety. To preserve this multi-modality and further encourage variation, we adopt a diffusion policy [44] as a generative prior over trajectories. We keep a history of observations and goals to predict the next  $H = 12$  horizon of the actions, while

only take the first action during inference time. By learning future steps of actions, the model can learn better transitions and relationship within histories of observations, and help predict the next-step action. Each observation and goals are embedded, with positional embedding and diffusion timestep embedding as well. A causal mask is used for attention computation, which means that action  $a_t$  in the horizon can only have access to the information up to time at  $t - 1$ .

2) *Adaptive Goal Mapping:* During training, key-frame goals are given and fixed according to the trajectories. However, in test time, the model cannot have access to which trajectory it needs to follow. Also, fixing goal sequences according to existing trajectories is not optimal especially in an environment that is different from training. To overcome this limitation, we introduce an online adapter as an MLP that dynamically adjusts key-frame conditions according to the current and past observations. This adapter will use a fixed-length history of observations to predict a feature vector which lies in the embedded space of the goal condition in diffusion models, as illustrated in Fig. 3. We performed normalization on this embedded space to make it a unit sphere. A key-frame feature codebook  $\mathcal{F} = \{f_{g_1}, f_{g_2}, \dots, f_{g_n}\}$ , with  $\|f_{g_j}\|_2 = 1$ , is pre-constructed from the augmented key-frames by passing into the fixed goal condition encoder in the diffusion model, where each entry stores the encoded feature  $f_{g_j}$ .

During inference, for every 5 steps of action, the adapter will predict a feature given the observations, and retrieve the most relevant feature  $f_g$  in the key-frame feature codebook by cosine similarity:  $j^* = \arg \max_j \frac{f_o^\top f_{g_j}}{\|f_o\|_2 \|f_{g_j}\|_2}$ , with  $f_g = f_{g_{j^*}}$ . By combining feature similarity with a normalized codebook on unit sphere, we ensure scale-invariant matching, preventing large feature norms from dominating and biasing reference selection. The selected feature  $f_g$  is then provided to the diffusion policy, replacing the static trajectory input with a context-aware reference. This retrieval process ensures that policy conditions adapt in real time to robot's state, while preserving safety through grounding in human priors.

## V. EXPERIMENTS

### A. Experimental Settings

**Implementations.** We trained our first-phase sparse-keyframe policy in IsaacGym [50]. The actor and critic network is a 2-layer MLP with hidden layer dimension [512, 256]. We trained in parallel with 4096 environments on Nvidia 4090 GPU for 5000 iterations per policy, which takes around 5 hours. We train our diffusion model for 1000 epochs, which takes around 40 hours on a single GPU of Nvidia A40. The robot we deploy on is 23-dof Unitree G1.

**Metrics.** We evaluate models based on three core criteria with levels of granularity: goal completeness, safeness, and efficiency. 1) For fall damage mitigation, we follow the evaluation criteria in [4], focusing on safeness with the use of peak instantaneous impulse on the base (PII), mean base acceleration (BA), and peak joint internal forces across all joints (PIF). Since damage in heavy robots most often

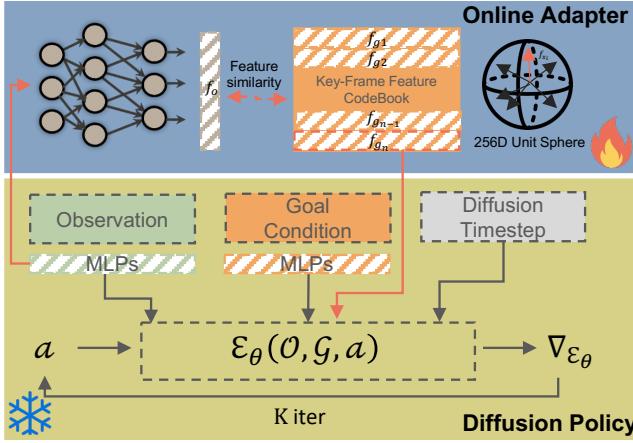


Fig. 3. **Overview of online adapter.** During inference, the adapter uses the history of observations to dynamically predict a feature and match with a key-frame goal feature in the code-book, and then pass the matched goal feature into the diffusion model to guide the process with context-awareness.

arises from high-impact stresses resulting from impulsive load transfer to the drives, higher values of these metrics indicate greater risk of damage and lower safety. 2) For fall recovery, we consider all three dimensions - a) *Goal Completeness*: measured by the success rate (SR, %), *i.e.*, the percentage of episodes where the robot's base height exceeds target height  $0.7m$  and the robot remains upright for a sustained duration; b) *Safeness*: measured by time-to-fall (TTF, seconds), which evaluates stability based on how long the robot can remain standing before another fall occurs. c) *Efficiency*: measured by the time-to-stand (TTS, seconds), evaluating the duration required for the robot to return to a stable height. Unless specified, simulated experiments are conducted with 512 randomly spawned robots over 7.5s on uneven terrains, with randomized initial fall configuration, base mass, and noisy observations. Results are averaged over 5 runs to minimize random biases and verify robustness.

#### B. Fall Damage Mitigation

**Settings.** As current research does not directly support fall damage mitigation on humanoid robot G1, we implement three baselines to compare with FIRM: 1) *freezing model*, where the robot output zero torque during falling, resulting in a passive collapse; 2) *dense keyframe tracking model*, where the robot follows dense keyframe references extracted from demonstrations; and 3) *sparse keyframe tracking policy*, where the robot follows sparse keyframe references extracted from demonstrations with only tracking rewards. This comparison setup allows us to investigate several key factors essential for fall damage mitigation: the passive vs active control strategies, the function of human demonstration, and the effect of adaptive memory for robust behaviors across diverse falling conditions.

**Results.** We observed several key observations from Fig 4(a) and Fig 4(b): 1) the freezing model yields the highest accelerations and joint forces, and its impulses are also the largest with multiple peaks (exceeding 400N), indicating that purely passive collapse exposes the robot to severe impact stresses;

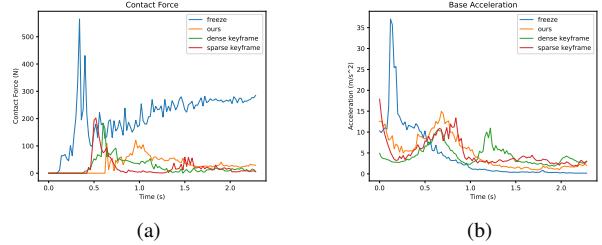


Fig. 4. (a) Distribution of contact force on the base over all time steps. Time steps with base contact impulse below 0.05Ns are not included. (b) Base acceleration (BA) during the fall.

TABLE II

**HUMANOID G1 ROBOT FALL RECOVERY RESULTS IN SIMULATED ENVIRONMENTS. COMPARISON BETWEEN HoST [8] AND OURS ACROSS THREE SCENES. N/A REFERS TO NO FAILURE CASES AS STANDING FIRST AND FALLING LATER.**

Terrain	Method	SR↑	TTF↑	TTS↓
Flat	HoST [8]	<b>99.40</b> ( $\pm 0.89$ )	0.06 ( $\pm 0.12$ )	<b>1.75</b> ( $\pm 0.03$ )
	<b>FIRM (Ours)</b>	96.29 ( $\pm 5.27$ )	N/A	2.47 ( $\pm 0.90$ )
Uneven	HoST [8]	23.20 ( $\pm 3.93$ )	1.87 ( $\pm 1.28$ )	3.10 ( $\pm 0.22$ )
	<b>FIRM (Ours)</b>	<b>93.20</b> ( $\pm 2.59$ )	<b>1.94</b> ( $\pm 0.95$ )	<b>2.86</b> ( $\pm 1.09$ )
Wave	HoST [8]	10.20 ( $\pm 2.68$ )	1.62 ( $\pm 1.04$ )	<b>2.08</b> ( $\pm 0.08$ )
	<b>FIRM (Ours)</b>	<b>55.86</b> ( $\pm 2.49$ )	<b>1.89</b> ( $\pm 1.06$ )	2.37 ( $\pm 1.12$ )

2) Sparse and dense keyframe tracking reduce impact forces compared to freezing model, with ours full FIRM model performing best in both base impulse and base acceleration. The underlying cause is that guiding the robot to follow human keyframes alone is brittle when the fall deviates from demonstrated trajectories. In contrast, the augmentation benefits from damage-reduction rewards, learning to emerge energy-dissipating poses, refine contact timing and force distribution beyond human priors, while online adaptation adaptively provides a safe, target goal that suits for current state to reference. Together, FIRM achieves smoother impact absorption and reduced joint stress.

#### C. Fall Recovery

**Settings.** FIRM is our final policy (*i.e.*, diffusion policy with keyframe codebook aware adapter). We compare FIRM with HoST [8], a recent SOTA method for humanoid standing-up control. HoST learns standing-up motions from scratch using RL with a multi-critic architecture and curriculum-based training, where a separate policy is trained for each terrain. We assess both methods from two perspectives: a) robustness to external disturbances introduced by additional payloads, and b) robustness to varying terrains. For fairness, we re-implement HoST [8] using its official codebase and evaluate it under same simulation setup as FIRM.

**Results.** Tab. II and Tab. III indicate several insights. For flat terrain, our results are comparable to HoST across all three metrics, and we observed in experiments that once FIRM stands up on flat terrain, it does not fall again; therefore, the TTF is reported as N/A. For challenging terrains, our performance substantially surpasses HoST. Specifically,

TABLE III  
SUCCESS RATE UNDER DIFFERENT PAYLOAD MASSES.

Method	10kg	12kg	15kg	20kg
HoST [8]	<b>78.40</b> ( $\pm 3.58$ )	61.00 ( $\pm 6.04$ )	35.60 ( $\pm 4.67$ )	5.00 ( $\pm 1.58$ )
<b>FIRM(Ours)</b>	75.60 ( $\pm 4.61$ )	<b>72.61</b> ( $\pm 4.83$ )	<b>60.20</b> ( $\pm 5.81$ )	<b>21.20</b> ( $\pm 7.19$ )

TABLE IV

ABLATION STUDY OF FALL DAMAGE REDUCTION & RECOVERY IN SIMULATION. PEAK INTERNAL FORCE (PIF, N).

Method	Damage Reduction		Recovery (Overall)		
	PIF $\downarrow$	SR $\uparrow$	TTF $\uparrow$	TTS $\downarrow$	
Dense Keyframe	42.38 ( $\pm 18.85$ )	84.19 ( $\pm 3.96$ )	1.51 ( $\pm 0.70$ )	3.09 ( $\pm 0.12$ )	
Sparse Keyframe	41.87 ( $\pm 21.04$ )	89.20 ( $\pm 1.92$ )	2.49 ( $\pm 1.17$ )	2.98 ( $\pm 0.08$ )	
+ Augmentation Policy	<b>41.01</b> ( $\pm 17.80$ )	93.20 ( $\pm 2.59$ )	1.94 ( $\pm 0.95$ )	2.86 ( $\pm 1.09$ )	
Diffusion w/o Adaptor	43.07 ( $\pm 18.24$ )	92.32 ( $\pm 2.33$ )	3.21 ( $\pm 1.02$ )	2.99 ( $\pm 0.67$ )	
<b>FIRM(Ours)</b>	41.23 ( $\pm 17.47$ )	<b>94.10</b> ( $\pm 2.17$ )	2.73 ( $\pm 1.42$ )	<b>2.41</b> ( $\pm 1.23$ )	

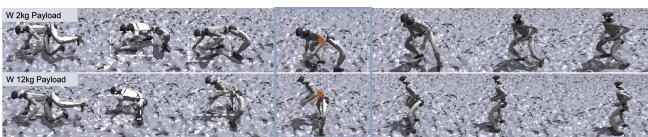


Fig. 5. Motion behaviors under different payloads. As the white boxes show, For 2 kg payload, the arms perform a “support-push” motion. For 12 kg, the robot’s arms make full contact with the ground, exhibiting a forceful pushing action to lift the body. The orange arrow indicates torso orientation.

FIRM achieves significantly higher success rates (70% improvement on uneven terrain and 45% on wave terrain), while also maintaining comparable TTS by one second on average. Moreover, HoST often fails completely when additional payloads are introduced, whereas FIRM maintains stable recoveries with only minor degradation. These results highlight that the adaptive keyframe memory and online goal remapping in FIRM are critical for scaling recovery behaviors beyond nominal training conditions.

#### D. Ablation Study

**Settings.** To analyze the contribution of each component in FIRM, we conduct ablation studies on three simplified variants: **1) Sparse Keyframe**, which directly tracks sparsely sampled human keyframes with only tracking rewards; **2) Sparse Keyframe + Augmentation Policy**, which augments demonstrations with RL and applies damage-reduction rewards; and **3) Diffusion w/o Adaptor**, which distills multimodal fall-recovery strategies into a diffusion model but lacks the adaptive observation-to-goal adaptor at inference. We compare these variants against our full method (**FIRM(Ours)**) under identical simulation settings, evaluating both fall damage reduction (PIF) and recovery performance (SR, TTF, TTS). Test terrains include *flat*, *uneven*, *wave*, and *rough*; among these, *uneven* terrain is included during training, while the others are unseen test conditions.

**Results.** The ablation results are shown in Tab IV. It highlights the importance of each design choice in FIRM. Using only human demonstrations provides a better baseline: the robot learns safer fall behaviors compared with freezing, but recovery success remains limited and brittle under out-of-

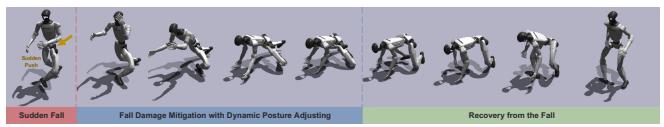


Fig. 6. Reaction to sudden fall. After a sudden extra force, the robot try to spread arms to avoid base collision and push arms and knees to recover.

distribution states. Incorporating reinforcement-based augmentation rewards significantly reduces peak impact forces and base accelerations, showing that contact timing and redistribution can be optimized beyond human priors. However, without adaptive conditioning, recovery trajectories are still restricted by the original demonstration modes. Finally, equipping the policy with the adaptive keyframe memory and online adapter yields the most substantial gains. By dynamically remapping goals according to current states, the adapter enables the diffusion policy to generalize across terrains and payloads, achieving highest success rates with smoother recoveries.

FIRM demonstrates superior ability to minimize impact forces through context-aware adaptation and to generalize robustly across diverse fall directions.

TABLE V  
SUCCESS RATE (%) FOR REAL-WORLD FALL RECOVERY ON THE UNITREE G1 ACROSS INDOOR TERRAINS (10 TRIALS EACH).

Terrain	G1 Controller	HoST [8]	<b>FIRM (Ours)</b>
Flat Mat	<b>10/10</b>	1/10	<b>10/10</b>
Slippery Surface	7/10	0/10	<b>8/10</b>

#### E. More Robustness Analysis

We test FIRM under various settings to show its robustness. Fig. 5 shows that under different payloads, FIRM adjusts its motion behaviors accordingly. Additional real world example can be seen at Fig. 8. With a sudden external force making the robot fall inevitably, the robot can mitigate falling and recover as seen in Fig. 6. Lastly, robot can regain balance successfully if a fall can be avoided (Fig. 7).

#### F. Real-world Comparison

**Settings.** We compare our FIRM against following baselines for fall recovery: **1) G1 Controller**, the default manufacturer-provided recovery controller on the Unitree G1, which executes a hard-coded joint trajectory with PD stabilization; and **2) HoST [8]**, the recent state-of-the-art learning-based approach that trains standing-up policies from scratch using RL. We evaluate all methods on two in-door test terrains: a flat mat matrix and a slippery surface created by placing plastic sheets over the mat, as we observe that even these relatively simple conditions pose significant challenges to the baselines. The evaluation metric is the success rate measured over 10 trials for each method and terrain.

**Results.** From Fig. 9 and Tab. V, we observe that: 1) HoST fails to consistently stand up across all terrains. From the qualitative snapshots, we observe that the robot’s legs often cross during the recovery motion under HoST policy, leading



Fig. 7. **Fall prevention.** Under 0.5s of zero torque output (mimics a sudden power outage in real world), the robot rapidly initiates rebalancing response, allowing the robot to maintain stability, thereby preventing a fall.

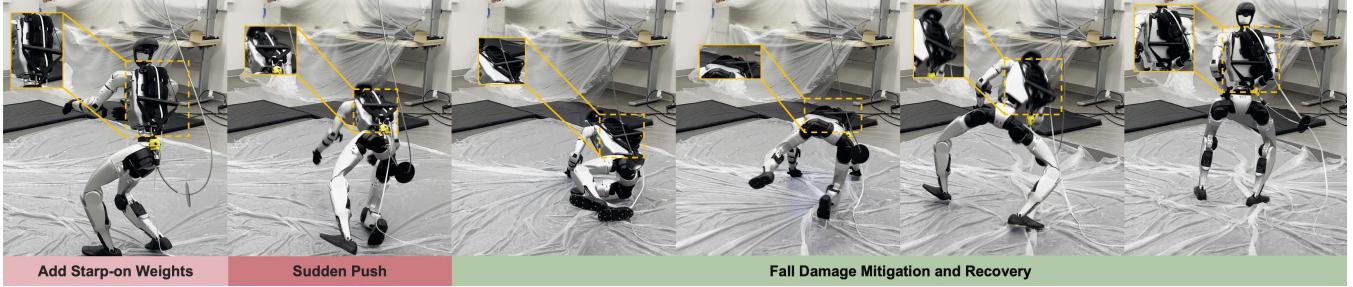


Fig. 8. **Fall recovery with deformable payload on FIRM.** The robot achieves robust fall recovery while carrying a 2.7kg payload on both *flat* and *slippery* terrains. As highlighted in the yellow boxes, the payload visibly swings and deforms, introducing additional disturbances and further demonstrating the stability of our approach.

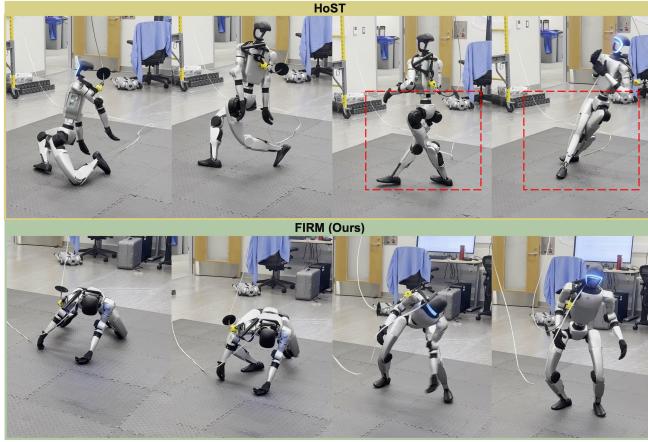


Fig. 9. **Comparison of fall recovery in real-world deployment.** While HoST [8] can barely stand, its leg movements are uncoordinated and the motion pattern departs from human-like behavior. Once upright, it fails to keep balance and quickly collapses. In comparison, our FIRM generates smoother and more natural motions, the robot uses arms to assist in a seamless fall-to-stand transition and sustain a stable posture after upright.

to instability and preventing the robot from achieving a stable standing posture. 2) The G1 controller can stand up on both terrains, but since it is equipped with only a single predefined posture, it cannot generalize to diverse falling configurations.

#### G. More Real-World Results

We provide additional real-world experiments in the *supplementary video* further to demonstrate the robustness of FIRM across diverse conditions, and investigate its limits. These include both indoor and outdoor terrains, variations in

payload, and scenarios requiring active balance maintenance.

## VI. CONCLUSIONS

We present FIRM, the first learning framework that unifies fall mitigation and recovery within a single humanoid control policy. FIRM explicitly balances safety and behavioral diversity, embedding both throughout the learning process. We highlight two key findings. **1)** Safety can be grounded in human priors - a few key human poses provide seeds for safe falling and rising; **2)** Generalization and adaptiveness emerge from reinforcement-augmented priors, post-stitching of trajectories, and an adaptive key-frame codebook memory, enabling responsive recovery across diverse disturbances. There remain two limitations. **1)** FIRM depends on nearest-neighbour matching in its key-frame codebook, which may limit performance in highly out-of-distribution scenarios. **2)** It also relies solely on proprioceptive data and therefore cannot yet exploit external cues from vision or tactile sensing to enhance environmental awareness and contact safety.

## REFERENCES

- [1] S. Kajita, F. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, “Biped walking pattern generation by using preview control of the zero-moment point,” in *ICRA*, 2003.
- [2] B. Stephens and C. G. Atkeson, “Push recovery by stepping for humanoid robots,” in *Humanoids*, 2007.
- [3] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, “Dynamic locomotion in the mit cheetah 3 through convex model-predictive control,” in *IROS*, 2018.
- [4] Y. Ma, F. Farshidian, and M. Hutter, “Learning arm-assisted fall damage reduction and recovery for legged mobile manipulators,” in *ICRA*, 2023.

[5] K. Fujiwara, F. Kanehiro, S. Kajita, K. Yokoi, H. Saito, K. Harada, K. Kaneko, and H. Hirukawa, “The first human-size humanoid that can fall over safely and stand-up again,” in *IROS*, 2003.

[6] L. Rossini, B. Henze, F. Braghin, and M. A. Roa, “Optimal trajectory for active safe falls in humanoid robots,” in *Humanoids*, 2019.

[7] S. Wang and K. Hauser, “Real-time stabilization of a falling humanoid robot using hand contact: An optimal control approach,” in *Humanoids*, 2017.

[8] 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.

[9] X. He, R. Dong, Z. Chen, and S. Gupta, “Learning getting-up policies for real-world humanoid robots,” in *RSS*, 2025.

[10] Z. Zhuang and H. Zhao, “Embrace collisions: Humanoid shadowing for deployable contact-agnostic motions,” *CoRR*, vol. abs/2502.01465, 2025.

[11] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, “Learning agile and dynamic motor skills for legged robots,” *Science Robotics*, 2019.

[12] Z. Chen, X. He, Y.-J. Wang, Q. Liao, Y. Ze, Z. Li, S. S. Sastry, J. Wu, K. Sreenath, S. Gupta *et al.*, “Learning smooth humanoid locomotion through lipschitz-constrained policies,” *arXiv:2410.11825*, 2024.

[13] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler, “Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters,” *ACM Transactions On Graphics (TOG)*, 2022.

[14] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” *IJRR*, 2023.

[15] M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim, “The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors,” in *Humanoids*, 2021.

[16] F. Liu, Z. Gu, Y. Cai, Z. Zhou, H. Jung, J. Jang, S. Zhao, S. Ha, Y. Chen, D. Xu *et al.*, “Opt2skill: Imitating dynamically-feasible whole-body trajectories for versatile humanoid loco-manipulation,” *arXiv:2409.20514*, 2024.

[17] 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)*, 2018.

[18] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi, “Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning,” in *CoRL*, 2024.

[19] T. Haarnoja, B. Moran, G. Lever, S. H. Huang, D. Tirumala, J. Humprik, 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*, 2024.

[20] Z. Zhuang, S. Yao, and H. Zhao, “Humanoid parkour learning,” in *CoRL*, 2024.

[21] K. Lin and S. X. Yu, “Let humanoids hike! integrative skill development on complex trails,” in *CVPR*, 2025.

[22] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang, “Expressive whole-body control for humanoid robots,” in *RSS*, 2024.

[23] X. Gu, Y.-J. Wang, X. Zhu, C. Shi, Y. Guo, Y. Liu, and J. Chen, “Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning,” in *RSS*, 2024.

[24] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, “Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control,” *IJRR*, 2025.

[25] I. Radosavovic, S. Kamat, T. Darrell, and J. Malik, “Learning humanoid locomotion over challenging terrain,” *arXiv:2410.03654*, 2024.

[26] K. Fujiwara, F. Kanehiro, S. Kajita, K. Kaneko, K. Yokoi, and H. Hirukawa, “Ukemi: Falling motion control to minimize damage to biped humanoid robot,” in *IROS*, 2002.

[27] K. Fujiwara, F. Kanehiro, S. Kajita, and H. Hirukawa, “Safe knee landing of human-size humanoid robot while falling forward,” in *IROS*, 2004.

[28] L. Meng, Z. Yu, W. Zhang, X. Chen, M. Ceccarelli, and Q. Huang, “A falling motion strategy for humanoids based on motion primitives of human falling,” in *International Conference on Robotics in Alpe-Adria Danube Region*, 2017.

[29] S.-H. Lee and A. Goswami, “Fall on backpack: Damage minimizing humanoid fall on targeted body segment using momentum control,” in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2011.

[30] J. Wang, E. C. Whitman, and M. Stilman, “Whole-body trajectory optimization for humanoid falling,” in *ACC*, 2012.

[31] S. Wang and K. Hauser, “Unified multi-contact fall mitigation planning for humanoids via contact transition tree optimization,” in *Humanoids*, 2018.

[32] R. Subburaman, J. Lee, D. G. Caldwell, and N. G. Tsagarakis, “Online falling-over control of humanoids exploiting energy shaping and distribution methods,” in *ICRA*, 2018.

[33] R. Subburaman, N. G. Tsagarakis, and J. Lee, “Online rolling motion generation for humanoid falls based on active energy control concepts,” in *Humanoids*, 2018.

[34] Z. Zhang, H. Liu, Z. Yu, X. Chen, Q. Huang, Q. Zhou, Z. Cai, X. Guo, and W. Zhang, “Biomimetic upper limb mechanism of humanoid robot for shock resistance based on viscoelasticity,” in *Humanoids*, 2017.

[35] D. Luo, Y. Deng, X. Han, and X. Wu, “Biped robot falling motion control with human-inspired active compliance,” in *IROS*, 2016.

[36] S.-k. Yun, A. Goswami, and Y. Sakagami, “Safe fall: Humanoid robot fall direction change through intelligent stepping and inertia shaping,” in *ICRA*, 2009.

[37] A. Goswami, S.-k. Yun, U. Nagarajan, S.-H. Lee, K. Yin, and S. Kalyanakrishnan, “Direction-changing fall control of humanoid robots: theory and experiments,” *Autonomous Robots*, 2014.

[38] J. Stückler, J. Schwenk, and S. Behnke, “Getting back on two feet: Reliable standing-up routines for a humanoid robot,” in *IAS*, 2006.

[39] F. Kanehiro, K. Fujiwara, H. Hirukawa, S. Nakaoka, and M. Morisawa, “Getting up motion planning using mahalanobis distance,” in *ICRA*, 2007.

[40] C. Yang, C. Pu, G. Xin, J. Zhang, and Z. Li, “Learning complex motor skills for legged robot fall recovery,” *RAL*, 2023.

[41] T. Tao, M. Wilson, R. Gou, and M. Van De Panne, “Learning to get up,” in *ACM SIGGRAPH 2022 conference proceedings*, 2022.

[42] A. Ajay, Y. Du, A. Gupta, J. B. Tenenbaum, T. S. Jaakkola, and P. Agrawal, “Is conditional generative modeling all you need for decision making?” in *ICLR*, 2023.

[43] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, “Planning with diffusion for flexible behavior synthesis,” in *ICML*, 2022.

[44] X. Huang, Y. Chi, R. Wang, Z. Li, X. B. Peng, S. Shao, B. Nikolic, and K. Sreenath, “Diffuseloco: Real-time legged locomotion control with diffusion from offline datasets,” in *CoRL*, 2025.

[45] X. Yuan, Z. Shang, Z. Wang, C. Wang, Z. Shan, M. Zhu, C. Bai, X. Li, W. Wan, and K. Harada, “Preference aligned diffusion planner for quadrupedal locomotion control,” *arXiv preprint arXiv:2410.13586*, 2024.

[46] G. Zhou, S. Swaminathan, R. V. Raju, J. S. Guntupalli, W. Lehrach, J. Ortiz, A. Dedieu, M. Lazaro-Gredilla, and K. P. Murphy, “Diffusion model predictive control,” *TMLR*, 2025.

[47] R. Römer, A. von Rohr, and A. Schoellig, “Diffusion predictive control with constraints,” in *7th Annual Learning for Dynamics & Control Conference*, 2025.

[48] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, “AMASS: Archive of motion capture as surface shapes,” in *ICCV*, 2019.

[49] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, “SMPL: a skinned multi-person linear model,” *ACM Trans. Graph.*, 2015.

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

[51] A. Allshire, H. Choi, J. Zhang, D. McAllister, A. Zhang, C. M. Kim, T. Darrell, P. Abbeel, J. Malik, and A. Kanazawa, “Visual imitation enables contextual humanoid control,” *arXiv:2505.03729*, 2025.

[52] Z. Luo, J. Cao, A. Winkler, K. Kitani, and W. Xu, “Perpetual humanoid control for real-time simulated avatars,” in *ICCV*, 2023.

[53] L. Yang, X. Huang, Z. Wu, A. Kanazawa, P. Abbeel, C. Sferrazza, C. K. Liu, R. Duan, and G. Shi, “Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction,” *CoRR*, 2025.

[54] S. Ross, G. Gordon, and D. Bagnell, “A reduction of imitation learning and structured prediction to no-regret online learning,” in *AISTATS*, 2011.

## APPENDIX

In this appendix, we provide: 1) additional implementation details (Sec. **A**); 2) benchmark details (Sec. **B**); 3) further experimental results and analysis (Sec. **C**); and 4) a discussion on the broader impact/limitations/future work of our method (Sec. **D**).

### A. Implementation Details

**Retargeting Human Videos.** For each recorded human motion video, we follow the pipeline of VideoMimic [51] to retarget human motions onto the G1 humanoid. The videos are captured in 4K at 60 fps and downsampled to 30 fps for processing. We found that the default configuration of VideoMimic often fails to preserve the spatial relationships between body parts, leading to inconsistent limb coordination and distorted postures. In particular, the hip pitch and yaw joints frequently exhibit excessive rotations exceeding 180°–360° relative to their neutral poses in our scenarios. This limitation arises because VideoMimic is primarily designed for low-contact motions (*e.g.*, walking, stepping, or sitting), whereas our sequences involve high-contact dynamics with frequent impacts during falls and recoveries. To mitigate this issue, we constrain the hip joint angle limits within physically plausible ranges that improve alignment between human and robot kinematics, resulting in more natural and structurally consistent retargeted motions.

There are also alternative frameworks to choose from, such as PHC [52], and OmniRetarget [53] *etc.*, imposing different physical constraints from different perspectives. However, our goal is not to achieve perfect physical fidelity, but to preserve the key motion intent in a physically reasonable manner, as we go beyond pure imitation that lacks contextual awareness, and instead use part of the retargeted motion sequence as a sparse prior. Residual inaccuracies and adaptation are then compensated through learning within the FIRM framework.

**Training Details.** All human demonstration videos are recorded on flat floors, while policy training for both stages (*i.e.*, skill priors, and adaptive memory learning) is trained on *uneven* terrains with domain randomization.

For trajectory collection used in distilling priors via the diffusion model, we record the following quantities: 1) root angular velocity  $\omega_{root}$ , 2) joint position and velocities  $q, \dot{q}$ , 3) last actions  $a_{t-1}$ . These quantities serve as the observation states for the diffusion policy, while the goal is represented by the joint positions of the corresponding target keyframe. For diffusion policy, we followed the transformer-based diffusion model as introduced in DiffuseLoco [44] with several architectural modifications: we employ a two-layer MLP to process the goal for enhancing its latent’s representational capacity. The goal-state MLP consists of two layers of 128 neurons each and outputs a 64-dimensional latent embedding. To maintain representational consistency, we use the same MLP in a Siamese manner to encode the current joint positions, and compute the difference between the goal-encoded and current-encoded latents. This Siamese design mimics the joint-position-difference formulation used

in our first-phase policy, allowing the diffusion model to reason about relative motion progress in latent space rather than relying solely on absolute joint configurations.

The online adapter is implemented as a three-layer 1D convolutional network that processes the past 50 timesteps along the temporal dimension. It predicts a latent representation mapped to the same feature space as the goal’s latent. The kernel sizes and strides for the three convolutional layers are [8, 4], [5, 1], and [5, 1], respectively. The output is normalized to lie on the unit sphere to ensure consistent feature magnitudes. The network is trained on the entire dataset for 20 epochs using a cosine similarity loss to optimize alignment between the predicted and target latents.

### B. Benchmark Details

**Test Environments.** While all human demonstrations are collected on flat floors, simulation training is performed on uneven terrains generated using Perlin noise. We test our approach in both simulation and real-world environments to evaluate robustness and generalization. 1) In simulation, we construct diverse terrains, including *flat*, *uneven*, *wave fields*, and *slopes*. We also design a special scenario with flat terrain and vertical walls to explicitly test fall-prevention behaviors when the robot falls toward obstacles. 2) In real-world experiments, we primarily test indoors with two levels of variation: friction and unevenness. For friction variation, we use different ground materials such as gym mats, plastic films, and protective wraps to create surfaces ranging from rough to slippery. For unevenness, we randomly scatter obstacles such as plastic foams and wooden planks to form irregular terrain patterns. Additionally, we evaluate outdoor performance on soft grass, earthen slopes, and sandy grounds resembling terrain on Mars, with varying inclines to assess stability and adaptability under natural conditions.

**Baselines.** Since no existing methods directly address the same task as ours, most of our comparisons are conducted against ablated variants of our own model. For the fall-recovery task specifically, we additionally compare against the default G1 controller, and HoST [8] that trains its policy purely through RL from scratch. We re-trained HoST following the official open-source implementation across multiple trials and observed notable discrepancies between our reproduced results and those reported in their paper. This issue was also discussed by other users as shown [here](#). As no further updates to the official code were available at the time we did the project, we used this retrained model for comparison. This outcome further highlights that learning fall recovery purely from scratch via reinforcement learning remains unstable and highly sensitive to reward design and environmental variations.

### C. More Results

**Why Human Prior Need to Be Sparse?** To analyze the effect of keyframe density during Stage-1 skill prior learning, we conduct an ablation study using different numbers of keyframes per video. We also include a comparison with the



Fig. 10. **Re-stand after a failed trial.** Instead of terminating upon a failed attempt, the online adapter dynamically re-evaluates the situation and identifies a new intermediate goal to initiate a subsequent re-stand trial.

TABLE VI

**ABLATION ON THE NUMBER OF KEYFRAMES.** SR: SUCCESS RATE (%), TTF: TIME-TO-FALL (S), TTS: TIME-TO-STEDY (S), PIF: PEAK INTERNAL FORCE (N).

# Keyframes	SR↑	TTF↑	TTS↓	PIF↓
Dense	84.19 ( $\pm 3.96$ )	1.51 ( $\pm 0.70$ )	3.09 ( $\pm 0.12$ )	42.38 ( $\pm 18.85$ )
75	87.19 ( $\pm 2.57$ )	0.62 ( $\pm 0.26$ )	3.02 ( $\pm 0.14$ )	41.98 ( $\pm 18.34$ )
50	88.60 ( $\pm 3.29$ )	2.92 ( $\pm 0.82$ )	<b>2.95</b> ( $\pm 0.13$ )	42.25 ( $\pm 20.92$ )
25	<b>89.20</b> ( $\pm 1.92$ )	2.49 ( $\pm 1.17$ )	2.98 ( $\pm 0.08$ )	41.87 ( $\pm 21.04$ )
10	86.40 ( $\pm 3.36$ )	2.16 ( $\pm 0.92$ )	3.33 ( $\pm 0.18$ )	<b>41.39</b> ( $\pm 21.02$ )

*Dense* keyframe setting, in which the policy follows interpolated motions between every frame, corresponding to an infinite number of keyframes in our human demonstrations. The test environments are kept identical to those described in Sec. V-D. Each human demonstration lasts approximately five seconds, and we evaluate keyframe frequencies of 15 Hz, 10 Hz, 5 Hz, and 2 Hz, respectively.

The results are shown in Tab VI. We observe that both the success rate (SR) and time to stand (TTS) improve as the number of keyframes decreases, but drop sharply when the count falls below 25. This trend can be attributed to two main factors: 1) the retargeted human demonstrations and preprocessing are not perfectly accurate, making precise motion tracking across dense keyframes infeasible; and 2) dense keyframe tracking lacks adaptability to terrain variations that differ from the original capture conditions. Using fewer keyframes allows the policy greater freedom to adapt its motions to new terrains while maintaining overall trajectory consistency. However, when the keyframe count becomes too sparse, the robot struggles to infer appropriate intermediate motions, leading to degraded performance. Thus, the human prior should be sparse yet structured, providing sufficient temporal relationship guidance while allowing flexibility for environmental adaptation. Based on this trade-off, we adopt 25 keyframes as the default setting for our algorithm.

**Supervised Finetuning, Naïve Distillation (MLP), or Diffusion?** There are multiple ways to integrate different skills or motion patterns into a unified policy. In this experiment, we evaluate two other major approaches. The first is *Supervised Finetuning*, where we train a policy on one motion (motion 1), then sequentially fine-tune it on subsequent motions (motion 2, motion 3, etc.). The second is *Distilling*, in which several expert policies are trained independently on different

motions and subsequently distilled into a single policy using DAgger [54]. To ensure a fair comparison, the distilled policy network in *Naïve Distillation (MLP)* shares the same MLP architecture as all expert and fine-tuned policies. For *Supervised Finetuning*, we train the initial motion for 5000 iterations and fine-tune on each subsequent motion for 3000 iterations. For *Naïve Distillation (MLP)*, we use DAgger to train the unified policy for 3000 iterations. Additionally, we include an *Expert Policy* baseline that employs the Sparse Keyframe with Augmentation setup described in Sec. V-D.

TABLE VII

**ABLATION ON THE APPROACH OF INTEGRATING MULTIPLE MOTIONS.** SR: SUCCESS RATE (%), TTF: TIME-TO-FALL (S), TTS: TIME-TO-STEDY (S), PIF: PEAK INTERNAL FORCE (N).

# Keyframes	SR↑	TTF↑	TTS↓	PIF↓
Expert Policy	<b>93.20</b> ( $\pm 2.59$ )	1.94 ( $\pm 0.95$ )	<b>2.86</b> ( $\pm 1.09$ )	<b>41.01</b> ( $\pm 17.80$ )
Supervised Finetuning	30.50 ( $\pm 15.20$ )	0.93 ( $\pm 0.20$ )	3.90 ( $\pm 1.12$ )	43.28 ( $\pm 17.32$ )
Distilling MLP	76.18 ( $\pm 5.23$ )	1.45 ( $\pm 0.87$ )	3.41 ( $\pm 0.94$ )	43.36 ( $\pm 18.21$ )
Diffusion w/o Adaptor	92.32 ( $\pm 2.33$ )	3.21 ( $\pm 1.02$ )	2.99 ( $\pm 0.67$ )	43.87 ( $\pm 18.24$ )

The results can be seen from Tab VII. We observe that the performance of *Supervised Finetuning* degrades significantly compared to *Policy Distillation (MLP)*. Sequential finetuning leads to catastrophic forgetting, causing the policy to retain only the most recently trained motion while losing earlier ones. While for *Distilling* with MLP, the performance also drops. We see that the model works well on relatively similar motions, while performing worse on others, causing the overall results to decrease. As the lengths for different motions are different, it is challenging for the phase term to capture the correct next-step dynamics for each, and the simple MLP architecture cannot effectively model the multimodal nature of these motions. These observations highlight the necessity of a diffusion-based approach, which can distill motion priors across heterogeneous demonstrations while preserving multimodality.

**The Importance of Online Adapter.** Online adaptation is crucial for a diffusion model to adapt in a dynamic environment where predefined motion sequences are infeasible. Rather than relying on a fixed trajectory, the online adapter serves as a motion planner that allows the robot to adjust its behavior according to the current state and surroundings. One significant outcome of our online adapter is the ability to re-stand from a previously “failed” sequence, demonstrating improved contextual awareness and adaptability. An example of such re-standing behavior is illustrated in Fig 10.

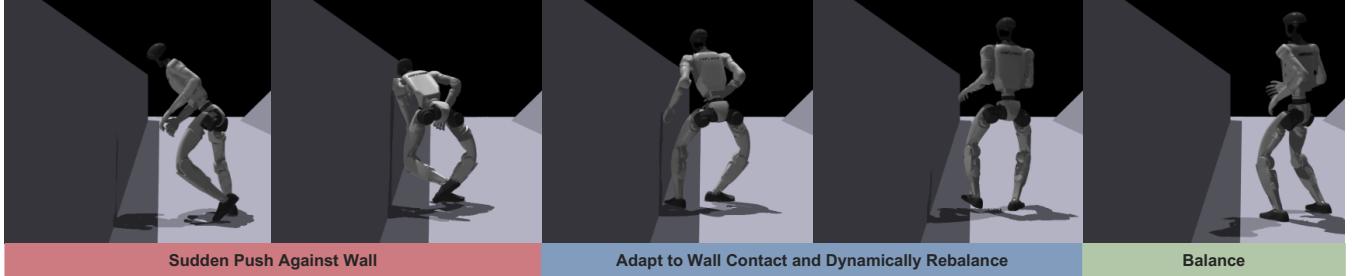


Fig. 11. **Fall prevention against wall.** Upon a sudden push toward the wall, the robot adapts to the contact, using it as support to stabilize its posture and avoid falling.

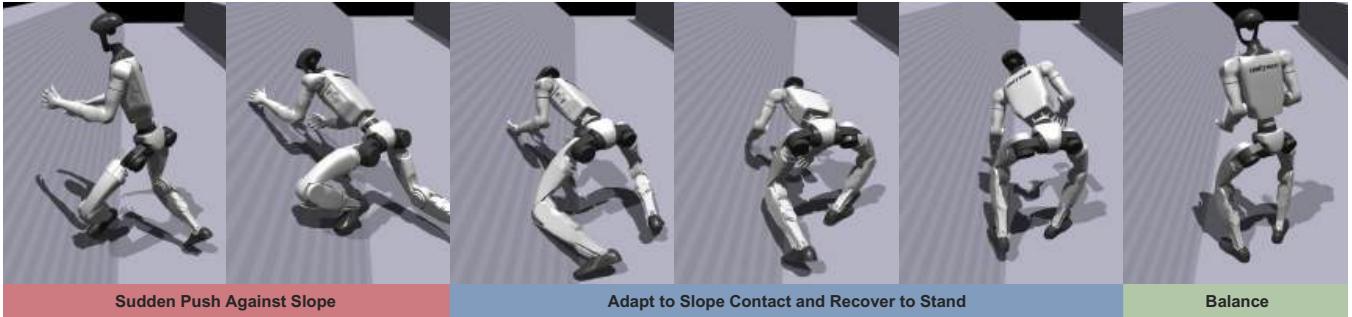


Fig. 12. **Fall prevention against slope.** Similar to the wall scenario, the robot adapts to the slope contact to prevent a full fall, and leverages the inclined surface as external support to recover and subsequently stand upright with regained balance.

**Additional Fall Prevention Results.** Additionally, we test the fall prevention performance against walls and slopes. Our robot demonstrates the ability to adapt to previously unseen environmental contacts, effectively leveraging surrounding structures to support itself and regain balance. Results in the simulation environment can be seen in Fig 11 and 12.

#### D. Discussion

FIRM goes beyond imitation learning or reinforcement learning alone, providing a viable pipeline for leveraging a small number of human demonstrations to solve complex, contact-rich tasks that are otherwise difficult to model through imitation or reward design to adapt in the dynamic environments. By learning from sparse human demonstrations, our framework can generate context-aware responses in time through the online adapter, adjusting its next goal to achieve stable task execution. Still, several limitations and open challenges remain: 1) As the number of our demonstrations is small and they are very iconic and different from each other, we rarely observe cases where the robot combines the falling phase of one motion with the early stage of recovery strategy of another sequence, more near the end when adjusting its final standing pose. This suggests that while our model achieves effective interpolation within each sparse motion segment in the learned goal feature space, the latent representations of different motion sequences remain relatively disjoint, limiting cross-motion composition. A promising direction for future work is to dig into more primitive and compositional motion structures, so that complex behaviors can emerge from more flexible recombination of simpler motor elements. Such a representation would enable smoother transitions and motion blending

between goal features, allowing the robot to dynamically compose strategies across diverse situations under just a few demonstrations. 2) Human priors should extend beyond motion trajectories to include decision-making processes, *i.e.*, understanding why humans choose specific recovery strategies under certain environmental conditions. Since our current demonstrations are recorded only on flat terrains, they involve limited decision complexity. Capturing first-person visual perspectives from humans performing falls across varied terrains could provide richer contextual cues, allowing robots to learn both perceptual grounding and adaptive decision-making in more dynamic environments. Integrating such perceptual and cognitive priors represents an exciting direction for future research.