Unified Humanoid Fall-Safety Policy from A Few Demonstrations

Zhengjie Xu[1], Ye Li[1], Kwan-Yee Lin[1] and Stella X. Yu[1,2]

***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 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.

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.

[1]University of Michigan, {zhengjie, yeyli, junyilin, stellayu}@umich.edu.
[2]S. Yu is also affiliated with UC Berkeley.

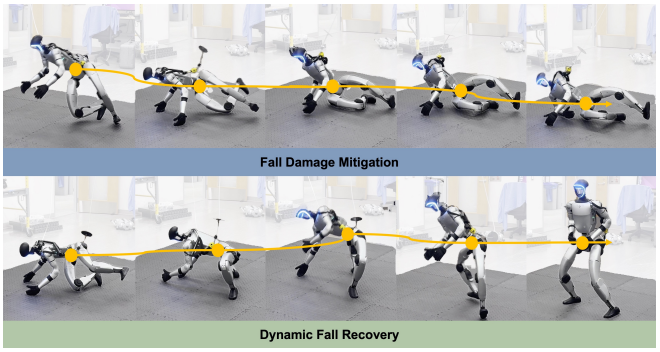


Fig. 1. **Our method enables humanoids to fall safely and rise promptly.** Snapshots show real-world deployment on the Unitree G1: When suddenly destabilized, the robot redirects into a side fall with arm buffering, then reorients and rises, demonstrating adaptive and resilient recovery.

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. By fusing sparse human demonstrations with reinforcement learning and an adaptive diffusion-based memory of safe reactions, we learn whole-body behaviors that cover fall prevention, impact mitigation, and rapid recovery within a single policy (Fig. 2).

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 maneuvers and diverse terrains [20]–[25]. Our work builds on 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.

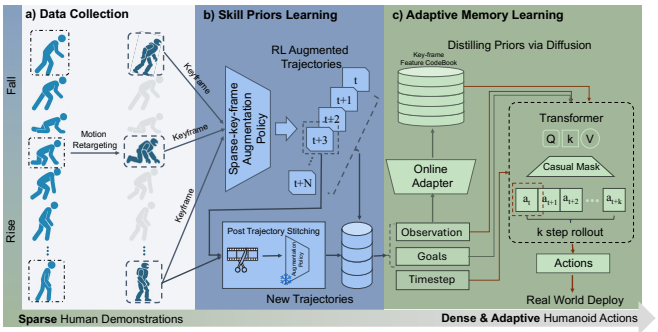


Fig. 2. **Workflow Overview**. From a few sparse human key poses, the robot seeds safe fall–rise skills, expands them through RL with post-trajectory stitching, distills enriched behaviors into a diffusion-based action memory, and composes online adapter to execute actions with context-awareness.

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, DiffuseLoco 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 but 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^{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 diversity*. 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: As safety 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 human video demonstrations, and process them through fitting to SMPL [49] and retargeting to the G1 humanoid via an off-the-shelf method VideoMimic [50]. In total, we use 4 high-quality trajectories covering forward, sideways, and backward fall-recover processes on flat ground, collected from varied subjects to provide diversity in motion styles.

While the data volume of demonstrations is small in scale, its quality and varieties are crucial, as they provide sufficient prototypes for our later designs to expand upon. Each trajectory includes joint positions q_i, rigid body poses relative to root T_{b_i} and root poses T_{root} retargetted to G1 robot. We calculate velocity and twists through 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 $H = 10\,\text{s}$, 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 $\mathscr{P} = \{P_1, P_2, \ldots, P_N, P_{\text{stand}}\}$, 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_{stand} until the episode terminates. For each trajectory, we train a corresponding policy.

State Initialization and Rollout. At the beginning of each episode, we randomly initialize the robot's state using a frame P_0 from the *dense* trajectory, while using only sparse key-frames for motion tracking. To simulate diverse fall conditions, we then randomize the robot states slightly and disable the actuators for a random duration $[0.04, 1.0]\,\text{s}$, 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. 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 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.

Policy Optimization. We optimize our policy using an asymmetric actor-critic framework with PPO [13], [17]. The critic has access to privileged information from the simulator that is unavailable to the actor. The actor's observation space consists of: 1) root angular velocity ω_{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 ϕ. The critic network additionally observes root linear velocity v_{root}. The phase ϕ is calculated by dividing the current runtime t by the length of the trajectory in time T and clipped to 1 for timesteps exceeding the trajectory length.

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 ($\mathscr{U}(0.25, 1.75)$), payload ($\mathscr{U}(-1, 1)$), and gains for each joint (p-gain:$\mathscr{U}(0.9, 1.1)$, D-gain: $\mathscr{U}(0.9, 1.1)$), 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 other works, episodes terminate when base height drops below a threshold or when collisions occur, to ensure the robot remains safe. These criteria does not fit, since we explicitly require the robot to fall to the ground and recover. Likewise, motion-tracking tasks [22] often use deviations from reference to terminate. As we only track *sparse* key-frames and aim to encourage exploration in between, this signal is also inappropriate. Therefore, our episode termination is kept minimal, while

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	$\sum_j \max(0, \lvert q_j - q_j^{\text{limit}} \rvert)$	-10
Joint velocity limit	$\sum_j \max(0, \lvert \dot{q}_j - \dot{q}_j^{\text{limit}} \rvert)$	-5
Action rate	$\sum (a[t] - a[t-1])^2$	$-1e^{-3}$
Torques	$\sum_j \tau_j^2$	$-1e^{-6}$
Acceleration	$\sum_j \ddot{q}_j^2$	$-2.5e^{-7}$
Body collision	$\sum_B \lVert \lambda_B \rVert^2$	$-1e^{-7}$
Momentum change	$\sum_B \lVert m_B a_B \rVert$	$-5e^{-3}$
Body yank	$\sum_B \lVert F_B \rVert^2$	$-2e^{-6}$

safety and stability are instead encouraged through reward.

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, 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*: we add penalizing rewards on body collision, momentum change, and body yank as described in [4] to mitigate the fall damage. The scale and 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 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 reuses 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: $G^{\text{new}} = \{(s_0,a_0),...,(s_t,a_t),(s_{t'},a_{t'}),...,\}$. 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. Since the distribution of these pairs is inherently multi-modal, directly distilling into 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 the diffusion policy [44]. 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 observation history. 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 $\mathscr{F} = \{f_{g_1}, f_{g_2}, \ldots, f_{g_n}\}, with \quad \lVert f_{g_j} \rVert_2 = 1$, 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 observation history, and retrieve the most relevant feature f_g in the key-frame feature codebook by cosine similarity: $j^* = \arg\max_j \frac{f_o^\top f_{g_j}}{\lVert f_o \rVert_2 \lVert f_{g_j} \rVert_2}, with \quad f_g = f_{g_{j^*}}$. 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.

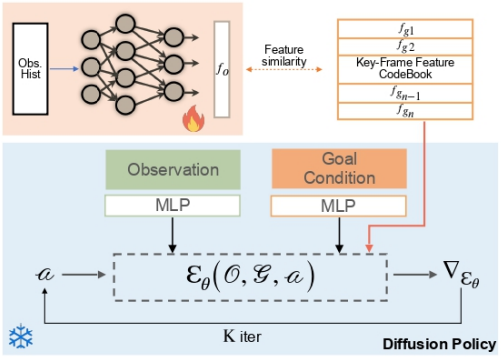


Fig. 3. **Overview of online adapter.** During inference, the adapter uses the history of observations to dynamically predict a feature and match with a key-frame goal feature in the code-book, and then pass the matched goal feature into the diffusion model to guide the process with context-awareness.

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

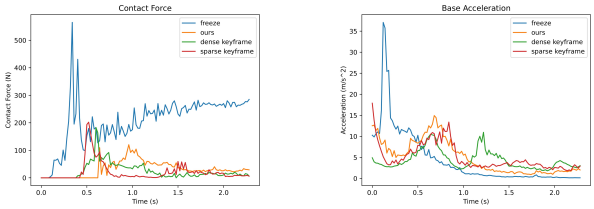
A. Experimental Settings

Implementations. We trained our part-1 sparse-key-frame policy in IsaacGym [51]. 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 robot.

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 mean base acceleration (BA), and peak joint internal forces across all joints (PIF). Higher values of these metrics indicate greater risk of damage and lower safety. **2)** For fall recovery, we consider all three dimensions. *Goal Completeness*: measured by the success rate (SR, %), the percentage of episodes where the robot's base height exceeds target height $0.7m$ and the robot remains upright for a sustained duration; *Safeness*: measured by time-to-fall (TTF, seconds), which evaluates stability based on how long the robot can remain standing before another fall occurs. *Efficiency*: measured by the time-to-stand (TTS, seconds), evaluating how long the robot returns 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.

B. Fall Damage Mitigation

Settings. As current work does not 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, resulting in a passive collapse; **2)** *dense keyframe tracking*, where the robot follows dense keyframe references extracted from demonstrations; and **3)** *sparse keyframe tracking*, where the robot follows sparse keyframe references 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 sparse prior, and the effect of adaptive memory for robust behaviors across diverse falling conditions.



(a) (b)

Fig. 4. (a) Distribution of contact impulse on the base (PII) over all time steps. Time steps with base contact impulse below 0.05 Ns 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]	**99.40** (±0.89)	0.06 (±0.12)	**1.75** (±0.03)
	FIRM (Ours)	96.29 (±5.27)	**N/A**	2.47 (±0.90)
Uneven	HoST [8]	23.20 (±3.93)	1.87 (±1.28)	3.10 (±0.22)
	FIRM (Ours)	**93.20** (±2.59)	**1.94** (±0.95)	**2.86** (±1.09)
Wave	HoST [8]	10.20 (±2.68)	1.62 (±1.04)	**2.08** (±0.08)
	FIRM (Ours)	**55.86** (±2.49)	**1.89** (±1.06)	2.37 (±1.12)

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; **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 key-frame 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

TABLE III

SUCCESS RATE UNDER DIFFERENT PAYLOAD MASSES.

Method	10kg	12kg	15kg	20kg
HoST [8]	**78.40** (±3.58)	61.00 (±6.04)	35.60 (±4.67)	5.00 (±1.58)
FIRM(Ours)	75.60 (±4.61)	**72.61** (±4.83)	**60.20** (±5.81)	**21.20** (±7.19)

TABLE IV

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

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

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, 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 key-frame 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 multi-modal fall-recovery strategies into a diffusion model without online adapter at inference. We compare these 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*; Only *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 reasonable baseline: the robot learns safer fall behaviors compared with freezing, but recovery success remains limited under out-of-distribution states. Incorporating reinforcement-based augmentation rewards significantly reduces peak impact forces and base accelerations, showing that contact 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 key-frame 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.

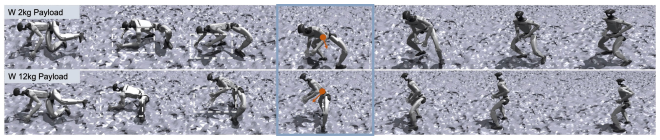


Fig. 5. **Adjustments under different payloads.** The orange arrow indicates torso orientation. For 2 kg payload, arms perform a "support–push" motion. For 12 kg, robot's arms make full contact with the ground to make a forceful pushing action to lift the body. This shows our policy can stay robust under OOD scenario.

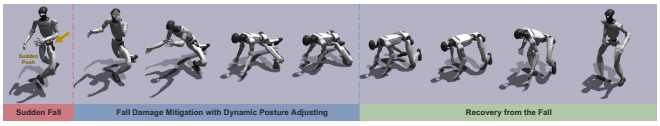


Fig. 6. **Natural reaction after falling by sudden external force.** For a scenario where it cannot maintain balance, the robot try to spread arms to avoid base collision and push arms and knees to recover. Note that the whole motion is very natural without abrupt actions.

Fig. 7. **Fall prevention when controllable.** 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. With stitching, robots are not necessary to fall every time it encounters imbalance situations.

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]	**FIRM (Ours)**
Flat Mat	**10/10**	1/10	**10/10**
Slippery Surface	7/10	0/10	**8/10**

E. 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], a 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 terrains: flat mat and slippery surface created by plastic sheets. We observe that even these relatively simple conditions pose significant challenges to the baselines. We evaluate the success rate

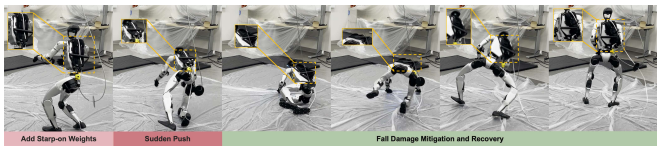


Fig. 8. **Fall recovery with deformable payload on FIRM.** As highlighted in the yellow boxes, the 2.7 kg payload visibly swings and deforms, introducing additional disturbances. The robot achieves robust fall recovery while carrying such payload on *slippery* terrains. With constant changes, this example demonstrates the stability of our approach.

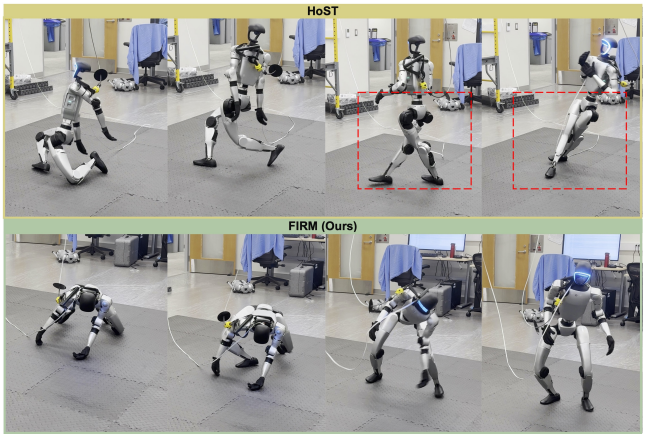


Fig. 9. **Comparison of fall recovery in real world.** HoST [8] can barely stand, with its leg movements uncoordinated and the motion pattern unnatural; Once upright, it fails to keep balance and quickly collapses. In comparison, our FIRM generates smoother and more natural motions, using arms to assist in a seamless transition and sustain stable after upright.

measured over 10 trials for each method and terrain.

F. 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).

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 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. Latent Analysis of Online Adapter

We provide additional t-SNE analysis for online adapter latent as shown in Fig 10, which illustrates the attribute of the adapter. 1) Though we only train with sparse human key-frame priors, our online adapter can learn a continuous manifold. 2) While some goal latent is learnt (blue oval), our policy might not achieve the exact pose, causing divergence from the goal latent. It internalizes to its own poses. 3) Though the state histories might diverge from each other as shown on the left t-SNE plot, online adpater can still map them to the same latent onto the right. Note that for the two light red ovals, as the right one, having the same frame which means the states are copied during initialization, is very far away from the left oval, they still map to similar latent.

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 exploit external cues from vision or tactile to enhance environmental awareness and contact safety.

Acknowledgments. We thank Osher Azulay, Utkarsh Singhal and Ryan Feng for their help with data collection. This project was supported, in part, by NSF 2215542, NSF 2313151, and Bosch gift funds to S. Yu at the University of Michigan and UC Berkeley, with additional compute support provided by the NAIRR Pilot under CIS240421, CIS250430, CIS240431.

References

[1] S. Kajita, F. Kanehiro, K. 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-agnostics 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.

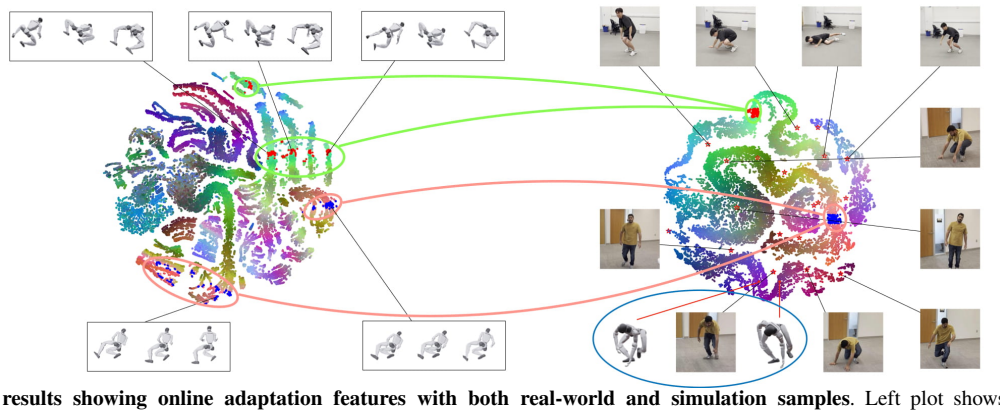


Fig. 10. **t-SNE results showing online adaptation features with both real-world and simulation samples**. Left plot shows 2D locations from proprioception history 2D t-SNE colored by inferred goal latent 3D t-SNE. As each state history contains 50 steps, the robot frames are at 0, 25, 50 timesteps. Right plot uses 2D t-SNE from goal latent and the same coloring as left plot.

[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. Humplik, 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] J. Long, J. Ren, M. Shi, Z. Wang, T. Huang, P. Luo, and J. Pang, "Learning humanoid locomotion with perceptive internal model," in *ICRA*, 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 a 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] 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.

[51] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Mack-

lin, D. Hoeller, N. Rudin *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv:2108.10470*, 2021.