

Test-Time Motion Steering for Perceptive Humanoid Fall Recovery

Osher Azulay*, Cheng-Lin Hsieh*, and Stella X. Yu

University of Michigan, Ann Arbor

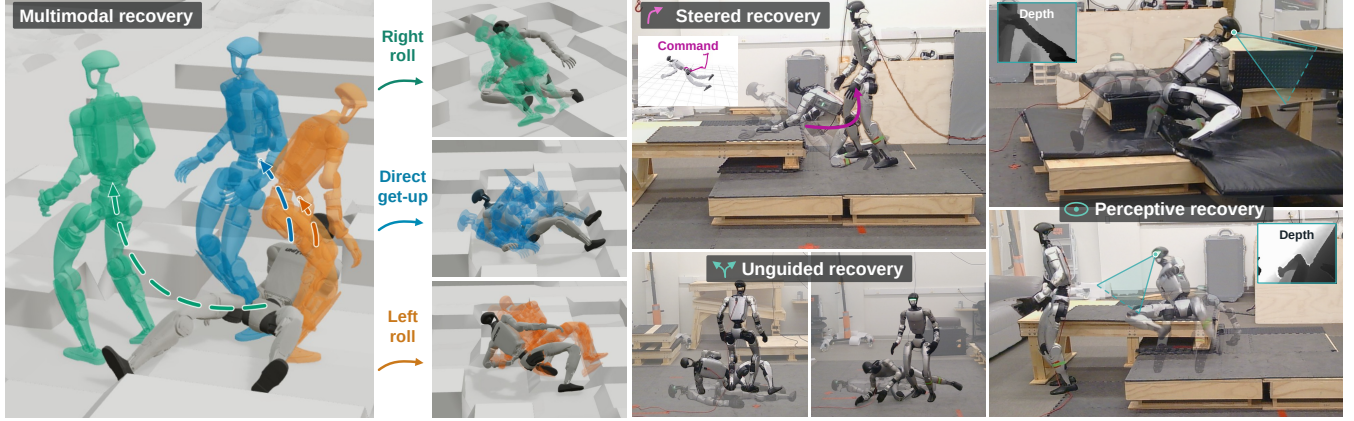


Fig. 1. **Perceptive and steerable humanoid fall recovery.** From the same initial pose, the planner generates alternative recovery motions, including right-roll, direct get-up, and left-roll recoveries (left). Hardware examples demonstrate steered and unguided recoveries (middle) and adaptation to different support configurations using visual observations (right).

Abstract—Humanoid fall recovery requires selecting a motion that suits both the surrounding scene and the task. The same fallen pose can admit several ways to stand, with different routes, support transitions, and final headings. Yet reliable recovery alone does not provide control over which of these alternatives the robot executes. Steering this choice requires anticipating the physical consequences of a requested movement while accounting for the surrounding scene. We present an approach to perceptive humanoid recovery that lets task objectives steer how the robot returns to standing at inference time. Our approach builds on two insights: diverse recoveries can be composed from shared movements, and anticipating how these movements unfold allows recovery to be steered toward a task objective. We learn a recovery vocabulary that connects intended movements to motor actions through sensory feedback. A visual diffusion planner learns from executions of this vocabulary to predict motion intent as future body states. Task costs guide this prediction toward a desired heading, rise profile, or route. The planner then generates motion latents conditioned on the steered intent, allowing the requested movement to shape its execution. In simulation, our approach recovers more reliably and with lower contact forces than prior controllers while keeping several routes available from the same pose. One trained model follows heading, rise-timing, and route objectives across terrains, and predicting states before motion latents makes this steering more precise and more stable. On a Unitree G1, the approach recovers in nearly all trials, follows commanded directions and routes, and uses depth to adapt recovery near a wall. Project page: [steerable-recovery](https://github.com/azulay/steerable-recovery).

I. INTRODUCTION

For a humanoid operating in cluttered environments, standing up is only part of recovering from a fall. A wall can obstruct a recovery route, a step can change the available

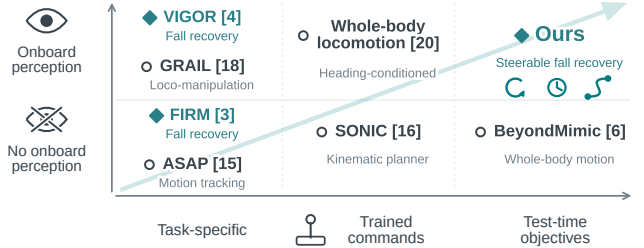


Fig. 2. **Scene awareness and motion steering.** Representative methods differ in their use of scene perception and how motion objectives are specified. **Diamonds** denote fall recovery; circles denote other whole-body motion. Our method steers recovery according to task objectives and the observed scene.

support, and the next task may require the robot to face a particular direction. Even from the same fallen pose, the robot may roll, turn on the ground, or pass through kneeling before rising. Recovery therefore requires choosing *how* to stand based on the surroundings and task (Fig. 1).

Recent controllers have made substantial progress in standing from diverse postures and managing falls [1]–[3]. Visual controllers further use terrain context to adapt recovery to uneven ground [4]. Building on this capacity to recover, we ask how task objectives specified at test time can guide the recovery motion (Fig. 2). Guided motion generation provides an interface for expressing such objectives [5], [6], but fall recovery cannot assume an upright starting pose or support through the feet. Realizing a requested movement therefore requires motor actions adapted to the robot’s fallen configuration and its contacts with the surrounding scene.

We express the movement to be steered as *motion intent*: a

*Equal contribution.

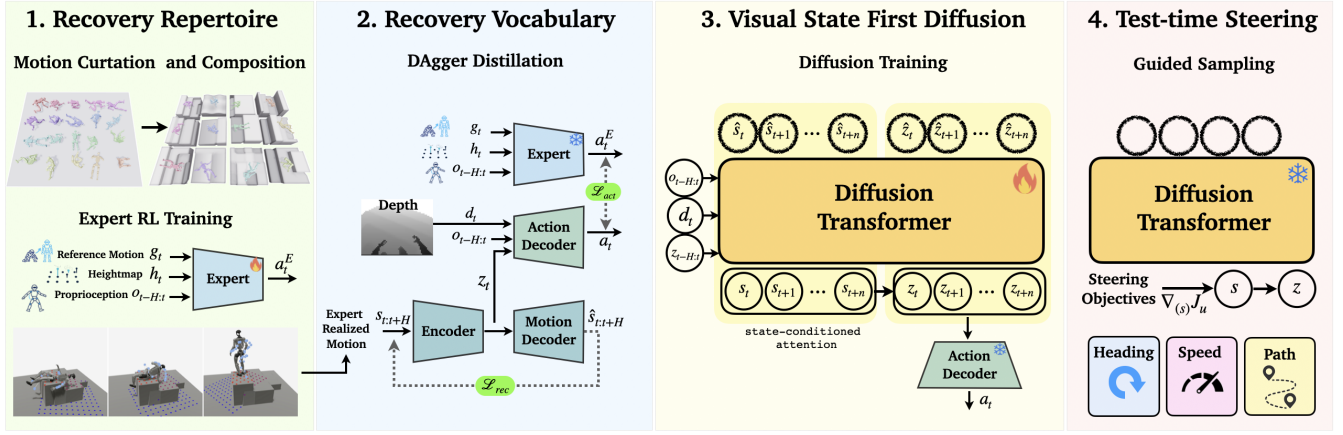


Fig. 3. **From recorded recoveries to guided execution.** (1) Composed motions become terrain-aligned references for a privileged expert. (2) An encoder maps expert-realized motion intent to motion latents; corrective motor-action queries train the decoder to execute that intent. (3) Selected decoder recoveries pair consumed motion latents with realized successor states for training the visual planner. (4) Task costs steer the predicted motion intent; this intent then conditions motion-latent generation, followed by feedback decoding into motor actions.

sequence of future body states.

Our first insight is that diverse recoveries can be composed from shared movement patterns. We compose compatible recorded segments [7], adapt them to terrain, and train an expert to realize these motions. From its executions, we learn a *recovery vocabulary*: compact motion latents that a planner can use to specify intended movements. A feedback decoder uses proprioception and depth to realize physically feasible recovery, adjusting motor actions to the current posture and terrain contacts.

Our second insight is that anticipating how recovery unfolds allows us to steer it toward a task objective. We train a visual diffusion planner on the decoder’s executions, pairing the motion latents used for control with the future body states actually reached (Fig. 3). These pairs connect each latent to the movement it produces under the observed posture and terrain. Task costs act on the predicted intent to steer heading, rise timing, or route without retraining for each objective. Whereas prior diffusion planners generate future states and execution commands jointly [6], [8], we factorize their generation: first steer the predicted body states, then generate motion latents conditioned on them. This makes the requested recovery an explicit condition for execution, while sensory feedback determines the motor actions needed under changing contacts.

Our contributions are threefold:

- 1) We introduce a framework for perceptive humanoid fall recovery that lets task objectives steer how the robot returns to standing. To our knowledge, this is the first framework to combine scene perception with test-time objective guidance for humanoid fall recovery.
- 2) We curate a multimodal fall-recovery dataset by composing recorded motions through motion matching, adapting them to terrain, and collecting their physical executions to capture diverse ways of returning to standing.
- 3) We introduce a visual state-first diffusion planner that learns from executed recoveries to anticipate their physical outcomes. Task objectives steer the predicted body

states before the planner generates motion latents conditioned on those states.

We evaluate the framework in simulation and on a Unitree G1. Compared with prior recovery controllers, it recovers more reliably and with lower contact forces, while keeping several routes to standing available from the same pose. The executed body motion closely follows the predicted intent, so task costs on this intent steer the actual recovery: one trained model follows heading, rise-timing, and route objectives across terrains. Depth helps in both the planner and the decoder, and predicting states first makes steering more precise and more stable than predicting states and motion latents jointly. On hardware, the robot recovers in nearly all trials, follows commanded directions and routes, and uses depth to adapt its recovery near a wall.

II. RELATED WORK

A. Humanoid Fall Recovery

Humanoid recovery requires coordinating the whole body through changing ground contacts. Early work designed protective falling and scripted get-up sequences [9], [10]. Learned controllers extended these capabilities to safe falling [11] and recovery from diverse postures [1], [2], [12]. FIRM unifies fall prevention, impact mitigation, and recovery using a small demonstration set [3], while VIGOR uses egocentric depth to adapt recovery to nearby terrain [4]. These methods emphasize reliable return to standing. We extend perceptive recovery to choosing among alternative ways to rise, using the observed surroundings and task objectives to guide direction, timing, and route.

B. Motion Priors and Reusable Motor Skills

Motion priors provide coordinated movements that can be reused across tasks. This view appears in studies of muscle synergies [13] and in motion graphs, which compose compatible recording segments into controllable sequences [7]. DeepMimic brings motion imitation into physical control, combining reference tracking with task objectives [14]. ASAP transfers motion-specific tracking poli-

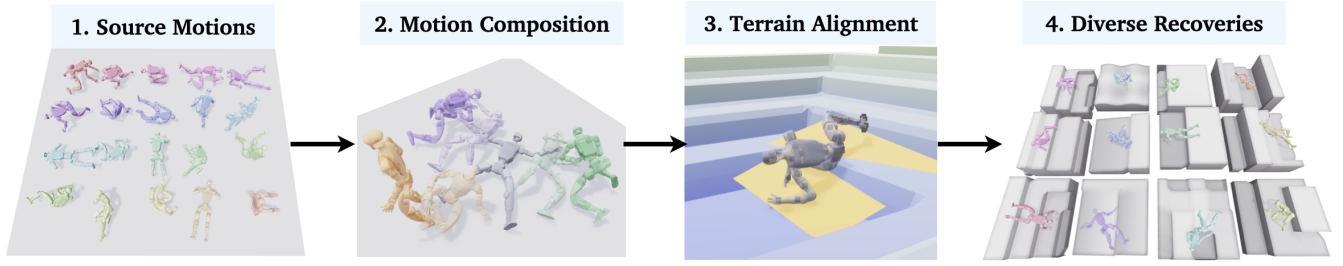


Fig. 4. **Curating a multimodal recovery set.** Source recovery motions (1) are composed through motion matching (2), aligned to terrain (3), and retained as diverse recovery references on terrain (4) for expert training.

cies to humanoid hardware by aligning simulated and real dynamics [15]. Learned motor representations extend this idea: SONIC provides a unified motion-token interface for whole-body control [16], and BFM-Zero supports behaviors specified through motion, goal, or reward prompts [17]. For interaction with the environment, GRAIL uses video priors and known scene geometry to construct aligned motion–scene data for learning visual loco-manipulation [18]. For recovery, reusable skills must accommodate changing ground contacts. We expand recorded motions through composition and terrain adaptation, then learn from their physical executions to provide alternative recoveries for visual planning.

C. Generative Motion Planning and Steering

Generative models allow task requests to shape future movement. DART generates motion from text and history and supports spatial guidance [5], while MaskedMimic uses motion inpainting to produce physical behavior from partial descriptions [19]. For humanoid locomotion, Zhang et al. [20] condition motion generation on heading commands and terrain observations. For control, a central issue is how generated body motion is connected to the actions that realize it. Decision Diffuser generates state trajectories and recovers actions through inverse dynamics [21]. Diffuse-CLoC jointly diffuses states and actions, using attention to condition actions on predicted states [22]. BeyondMimic jointly models states and motion latents, allowing task costs to guide motion that a learned controller executes [6]. We extend this approach to recovery conditioned on egocentric depth. Our state-first formulation steers predicted body motion before generating its motion latents, so control is conditioned on the selected recovery route.

III. METHOD

Our method lets a fallen humanoid choose among different ways to get up and steer its recovery toward a requested heading, route, or rise profile (Fig. 3). We first curate a *multimodal recovery repertoire* of different ways to get up. We then learn a *recovery vocabulary* and a decoder that converts the desired movement into motor actions using current observations. To *plan executable recovery*, we train a planner on the robot’s resulting motions. Finally, *steering recovery with motion objectives* lets a task shape the predicted motion intent and the motor actions generated to realize it.

A. Curating a Multimodal Recovery Repertoire

The multimodal recovery repertoire supplies examples for learning the recovery vocabulary (Fig. 4). We compose alternative recovery references, adapt them to terrain, and use a teacher controller to verify that they can be physically executed.

Motion composition. We create different ways to get up by combining compatible parts of recorded recoveries. Our sources are LAFAN1 [23], BONES-SEED [24], AMASS [25], and our seed recordings in G1 coordinates. Motion matching [7] operates at the frame level, while composition joins variable-length recorded segments. For an outgoing frame i , we select a compatible frame from another recording, $j^* = \arg \min_{j \in \mathcal{C}(i)} \|\mathbf{f}_i - \mathbf{f}_j\|_2^2$, where $\mathbf{f}$ contains normalized pose, velocity, body-attitude, and limb-proximity features, and $\mathcal{C}(i)$ restricts candidates to compatible frames near motion-phase boundaries. We align the recordings in position and heading, blend the junction, and continue along the incoming recording. Segment lengths follow the selected junctions rather than a fixed duration. Retaining several compatible matches creates alternative recovery sequences.

Terrain adaptation and expert realization. We align each reference with the ground to provide plausible support and clearance for learning physical recovery. Building on terrain-based vertical alignment [4], [19], we add a local support-plane rotation that tilts the fallen body and fades during the rise. At each frame, we transform reference body points as

$$\tilde{\mathbf{p}} = \mathbf{R}(\mathbf{p} - \mathbf{c}) + \mathbf{c} + \Delta z \mathbf{e}_z, \quad (1)$$

where $\mathbf{p}$ and $\tilde{\mathbf{p}}$ are the original and aligned points, $\mathbf{c}$ is the fitted plane’s center, $\mathbf{R}$ provides the tilt, and Δz adds clearance along the vertical unit vector $\mathbf{e}_z$. Geometric checks reject invalid support, penetration, and discontinuities, yielding tasks comprising terrain, an initial pose, and coarse motion intent.

Expert training. A teacher policy π_E learns to turn these references into physical recoveries. Following prior work that combines motion priors with terrain-aware control [4], coarse references guide the recovery maneuver while reinforcement learning adapts its execution to terrain and contact. We train the expert with PPO [26] to output joint targets for PD control, $\mathbf{a}_t^E = \pi_E(\mathbf{o}_{t-H+1:t}, \mathbf{g}_t, \mathbf{h}_t)$, where $\mathbf{o}_{t-H+1:t}$ is a history of H proprioceptive observations, $\mathbf{g}_t$ contains sparse reference targets, and $\mathbf{h}_t$ contains privileged local terrain heights. Each observation $\mathbf{o}_t$ includes joint positions and

velocities, body angular velocity, gravity direction, and the previous motor action. Expert execution verifies feasibility and pairs realized recoveries with their coarse references.

B. Learning a Recovery Vocabulary

The recovery vocabulary connects short movements from the teacher’s recoveries to a decoder that adjusts its motor actions to the robot’s current posture and nearby terrain.

Motion representation. We describe the intended movement as a short sequence of future body states, called motion intent. Each state contains joint configuration and pelvis motion:

$$\mathbf{s} = [\mathbf{q} - \mathbf{q}_{\text{default}}, \dot{\mathbf{q}}, \Delta \mathbf{p}, \mathbf{r}], \quad (2)$$

where $\mathbf{q}$ and $\dot{\mathbf{q}}$ are joint angles and velocities, $\mathbf{q}_{\text{default}}$ is the default joint configuration, $\Delta \mathbf{p}$ is pelvis displacement, and $\mathbf{r}$ stores the first two columns of the pelvis rotation matrix. We express recorded expert poses relative to the learner’s current pelvis:

$$\begin{aligned} \Delta \mathbf{p}_{t+i} &= (\mathbf{R}_t^h)^\top (\mathbf{p}_{t+i}^{\text{rec}} - \mathbf{p}_t), \\ \mathbf{r}_{t+i} &= \text{Rot6D}((\mathbf{R}_t^h)^\top \mathbf{R}_{t+i}^{\text{rec}}), \end{aligned} \quad (3)$$

where t is the current time and i is the sample offset. The superscript *rec* denotes recorded expert poses; $\mathbf{p}_t$ and $\mathbf{R}_t^h$ are the learner’s current pelvis position and gravity-aligned heading frame. The operator *Rot6D* extracts the first two rotation-matrix columns. This common anchor preserves tracking error across the window. Causal heading transport through body rotations maintains the frame during lying and rolling.

Intent encoding and feedback. An encoder compresses the desired state window into a motion latent. The decoder combines it with sensory observations to produce motor actions:

$$\mathbf{z}_t = \mathcal{E}(\mathbf{s}_{t+\mathcal{I}}, \mathbf{m}), \quad (4)$$

$$\hat{\mathbf{a}}_t = \pi(\mathbf{o}_{t-H+1:t}, \mathbf{z}_t, \mathbf{d}_t), \quad (5)$$

where $\mathcal{I}$ is the set of sampling offsets, $\mathbf{m}$ marks available samples, $\mathbf{z}_t$ is the motion latent, $\hat{\mathbf{a}}_t$ is the predicted motor action, and $\mathbf{d}_t$ denotes egocentric depth observations. Proprioception enables correction, while depth conditions motor actions on nearby terrain.

Imitation and reconstruction. Desired state windows come from recorded teacher executions, re-expressed relative to the learner’s current pelvis position and heading using (3).

We train the decoder on corrections to the robot’s own movements using online DAgger [27]. Corrective teacher motor actions $\mathbf{a}_t^E$ are queried at learner-visited states, using the recording’s original coarse reference at the corresponding reference time. Learner-induced deviations create varied intent–state pairings, training the decoder to adapt recorded intent to states encountered during its own execution.

To encourage motion latents to retain the intended movement, we reconstruct the desired state window from the latent alone: $\hat{\mathbf{s}}_{t+\mathcal{I}} = \mathcal{R}(\mathbf{z}_t)$, where $\mathcal{R}$ is a training-only reconstruction head. The combined loss is

$$\mathcal{L} = \text{MSE}(\hat{\mathbf{a}}_t, \mathbf{a}_t^E) + \lambda \mathcal{L}_{\text{rec}}, \quad (6)$$

where $\mathcal{L}_{\text{rec}}$ is the weighted mean squared error between

reconstructed and desired states over valid samples, and λ sets its weight. Reconstruction balances normalized feature groups and emphasizes near-future samples.

C. Learning to Plan Executable Recovery

Offline training data. The planner learns from the robot’s realized motion under varied physical and terrain conditions. We freeze the learned encoder and decoder and collect rollouts using recorded motion intent across the curated tasks. Pairing each motion latent used for control with the resulting next state provides offline training sequences for the planner.

Factorized, state-first diffusion. The planner predicts future body states before generating the motion latents to realize them. Both predictions use the context

$$\mathbf{c}_t = (\mathbf{o}_{t-H+1:t}, \mathbf{z}_{t-H_z:t-1}, \mathbf{d}_t), \quad (7)$$

where H_z is the number of preceding motion latents. Future states share the current anchor in (3). To steer motion intent while coupling it to execution, we factor

$$p(\mathbf{S}, \mathbf{Z} | \mathbf{c}_t) = p(\mathbf{S} | \mathbf{c}_t) p(\mathbf{Z} | \mathbf{S}, \mathbf{c}_t), \quad (8)$$

where $\mathbf{S} = \mathbf{s}_{t+1:t+F}$ contains the predicted states and $\mathbf{Z} = \mathbf{z}_{t:t+F-1}$ their associated motion latents over F future steps. Steering $\mathbf{S}$ changes the motion intent that conditions latent generation.

A shared Transformer implements both conditional distributions with diffusion [28]. State and latent tokens use separate input projections and output heads. Directed attention allows state tokens to attend to context and other states, while latent tokens additionally attend to the complete state sequence. At inference, we first denoise future states from Gaussian noise, then hold the resulting sequence fixed while denoising the motion latents that realize it.

We train on paired latent sequences and their realized state trajectories, sampling state prediction and conditional latent prediction with equal probability. For state prediction, future states are corrupted and future latents are masked; for latent prediction, latents are corrupted and the corresponding states provide conditioning. Both tasks predict clean targets using weighted mean-squared error. Small, temporally correlated perturbations of the conditioning states encourage tolerance to prediction errors at deployment. Training requires no task-command labels.

D. Steering Recovery with Motion Objectives

We steer recovery at test time by applying task costs to predicted motion intent.

Steering motion intent. A task defines a differentiable cost on motion intent, enabling test-time steering without task-specific training pairs. During DDIM sampling [29], we evaluate this cost on the denoised prediction and backpropagate through the denoiser to obtain the guidance correction:

$$\Delta \mathbf{x} = -\eta \nabla_{\mathbf{x}} J(\mathbf{S}(\mathbf{x})), \quad (9)$$

where $\mathbf{x}$ is the noisy state sequence, $\mathbf{S}(\mathbf{x})$ is its denoised prediction, J is the task cost, and η is the effective guidance step size. This update shifts sampling toward lower task

cost while the learned diffusion prior supplies the recovery repertoire.

From intent to execution. We hold the steered motion intent $\mathbf{S}^*$ fixed while sampling $\mathbf{Z} \sim p(\mathbf{Z} | \mathbf{S}^*, \mathbf{c}_t)$, then decode these latents into motor actions using current observations (Fig. 3). We execute a short prefix and replan from updated observations.

We illustrate this interface with three objectives that steer recovery heading, rise timing, and pelvis path.

Heading steering. We steer recovery toward a desired heading by penalizing the wrapped angular error:

$$J_{\text{heading}} = \text{wrap}(\alpha_F - \alpha^*)^2, \quad (10)$$

where α_F is the predicted heading at the last of F future steps, α^* is the target heading, and wrap maps the angular difference to $[-\pi, \pi)$.

Rise timing. Following a desired height profile encourages the robot to rise earlier or later:

$$J_{\text{rise}} = \frac{1}{F} \sum_{j=1}^F (h_j - h_j^*)^2, \quad (11)$$

where h_j and h_j^* are the predicted and desired pelvis heights at future step j , both relative to the initial pelvis height.

Pelvis-path steering. We guide the pelvis route and posture by matching target poses:

$$J_{\text{path}} = \frac{1}{F} \sum_{j=1}^F [\|\mathbf{p}_j - \mathbf{p}_j^*\|_2^2 + \lambda_R \|\mathbf{R}_j - \mathbf{R}_j^*\|_F^2], \quad (12)$$

where $\mathbf{p}_j$ and $\mathbf{R}_j$ are the predicted pelvis position and rotation matrix at step j , and stars mark their targets. The weight λ_R balances orientation against position; $\|\cdot\|_F$ denotes the Frobenius norm.

IV. EXPERIMENTS

We first test whether recovery is reliable and safe while keeping several routes to standing available (Sec. IV-B). We then examine whether the robot executes the predicted motion intent (Sec. IV-C) and whether task costs can steer recovery with one trained model (Sec. IV-D). Ablations examine depth perception and prediction order (Sec. IV-E), and Sec. IV-F reports hardware experiments.

A. Setup and Metrics

Simulation setup. We evaluate the 29-DoF Unitree G1 in Isaac Sim using our visual diffusion planner and frozen depth-conditioned decoder (Sec. III-C). Unless noted otherwise, we sample a one-second motion intent and its motion latents at 50 Hz using eight DDIM steps per pass, execute the first 0.2 s, and replan with fresh noise. Attempts last 15 s. No external pushes are applied.

Environments and starts. All evaluations use terrain sites and starting configurations (*starts*) held out from training. The held-out set covers six terrain types, including stepped terrain, uneven support, abrupt drops, and inclines with height discontinuities (Fig. 5), and a wide range of initial body configurations.

Baselines. We compare against two recent recovery controllers, VIGOR [4] and FIRM [3], using checkpoints pro-

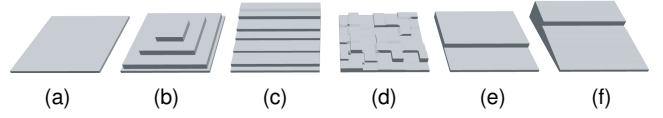


Fig. 5. **Evaluation terrains.** (a) Flat, (b) stepped pyramid, (c) stairs, (d) rubble, (e) drop edge, and (f) slope with a riser.

TABLE I
EVALUATION METRICS

Metric	Definition
Confirmed rise (%)	Pelvis at least 55 cm above the terrain and upright for 1 s, without falling back.
SR (%)	Stable recovery: a confirmed rise, still standing with low pelvis speed in the final 1 s.
Pelvis displ. (m)	Horizontal pelvis distance from its starting position at 15 s, averaged over all attempts.
Span (cm, deg)	Within-start 10th–90th percentile spread of forward, lateral, and heading change before rising.
Prediction error	Heading MAE (deg); height and 3D-position RMSE (cm) between predicted and executed motion.
Steering error	Heading MAE at confirmed rise (deg); rise-time MAE (s); height-profile and pelvis-path RMSE (cm).
Peak force (kN)	Largest net contact force on any link.
Peak torque (N m)	Largest absolute motor torque.
Saturation (%)	Joint-time samples at $\geq 99\%$ of the torque limit.
Joint-limit count	Most joints past their limits at once by >1 mrad.
Limit excess (mrad)	Largest excursion beyond a joint limit.

h : pelvis height above the local terrain; u : dot product of the pelvis and world up axes. A confirmed rise holds $h \geq 55$ cm and $u \geq 0.75$ for 1 s, then a 0.5 s follow-through, with no deep re-fall: a pelvis drop of at least 15 cm with $h < 25$ cm, or with $h < 35$ cm and $u < 0.5$, for 0.1 s. SR also excludes later deep re-falls and requires $h \geq 55$ cm and $u \geq 0.90$ in the final 1 s, with pelvis linear and angular speeds at most 0.75 m/s and 1.5 rad/s in at least 70% of samples. Spans use starts with at least ten confirmed rises. MAE: mean absolute error; RMSE: root mean squared error.

vided by their authors. VIGOR acts deterministically, and FIRM samples from its diffusion policy without added exploration noise. All methods use the same starts, recovery criterion, and time limit.

Metrics. Table I defines the metrics and their thresholds. We report two success rates: a *confirmed rise* requires the robot to stand up, and *stable recovery* (SR) also requires it to remain standing until the attempt ends. Success, safety, and effort metrics include all attempts. Spans, heading errors, and timing errors use trials with a confirmed rise; height and path errors include all attempts.

B. Reliable, Safe, and Diverse Recovery

Steering is useful only if the underlying recovery is reliable and leaves several ways to stand. We compare all methods over 920 trials per method across terrains. Besides success and the loads on the robot and terrain, we measure route spans: how far successful routes spread across repeated trials from the same start (Table II).

Recovery is more reliable and more controlled. FIRM, which has no terrain perception, rarely rises on these starts (7.8% confirmed rise), while VIGOR recovers in most attempts. Our method improves on VIGOR in both success and safety: it recovers more often (93.0% vs. 80.8% SR) and nearly halves the peak contact force (Table II). Lower torques and fewer joint-limit violations follow the same trend. Our

TABLE II

HIGHEST SUCCESS, LOWEST LOADS, AND WIDEST ROUTE SPANS

Metric	FIRM [3]	VIGOR [4]	Ours
<i>Recovery</i>			
Confirmed rise (%)	7.8	89.0	94.7
SR (%)	0.0	80.8	93.0
<i>Safety and effort (lower is better)</i>			
Peak force (kN)	7.55	2.50	1.37
Peak torque (N m)	126.4	101.3	83.3
Saturation (%)	8.57	0.34	0.30
Joint-limit count	0.55	0.58	0.20
Limit excess (mrad)	3.64	1.65	1.44
<i>Route span from the same start</i>			
Forward (cm)	23.6	0.0	32.9
Lateral (cm)	9.6	0.0	44.6
Heading (deg)	21.5	0.0	107.8

920 trials per method across terrains.

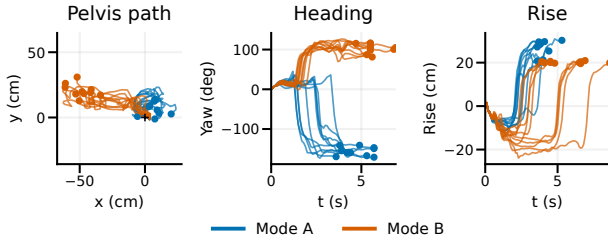


Fig. 6. **Distinct recovery routes from fixed starts.** Successful trials shown as pelvis path, heading, and pelvis rise. Colors mark two recurring recovery groups, with ten trials per group; dots mark confirmed rise.

method thus recovers more often and with lower loads, giving steering a controlled recovery to build on.

The learned prior keeps several routes available.

Repeated trials from the same start follow distinct routes rather than timing variations of one motion. Routes from a single start spread by about 110° in heading and more than 30 cm in forward and lateral displacement (Table II), and Fig. 6 shows distinct pelvis paths, turning directions, and rise profiles across repeated successful recoveries. VIGOR, a deterministic policy, repeats one route per start, and FIRM’s routes vary less and rarely end in standing. Our prior thus offers several reliable routes for steering to choose from.

C. Faithful Execution of Predicted Motion Intent

To steer these alternatives, the planner must predict how they will be executed. We compare predicted and executed motion on flat and uneven terrain. The robot executes the first 0.4 s of each prediction before replanning, and we compare pelvis, hands, and soles at matching times.

Predicted body motion is a reliable handle for steering.

Executed pelvis motion stays within a few degrees and a few centimeters of the prediction, even on uneven terrain (Table III). Hands and soles deviate about twice as much, because the decoder adapts contacts to the local support rather than reproducing the predicted limb positions. The prediction is thus accurate for the body but not for exact contacts. We therefore steer pelvis heading, height, and path, and leave contact placement to the decoder.

TABLE III

EXECUTED PELVIS MOTION CLOSELY FOLLOWS THE PREDICTION

Terrain	Pelvis			Limbs (L/R)	
	Heading (deg)	Height (cm)	Position (cm)	Hands (cm)	Soles (cm)
Flat	3.6	1.5	3.1	6.1/5.8	6.4/6.5
Uneven	6.0	2.1	4.6	9.0/9.1	9.5/9.8

64 attempts per terrain, using 0.4 s prediction windows before the rise. Heading is MAE; height and 3D positions are RMSE.

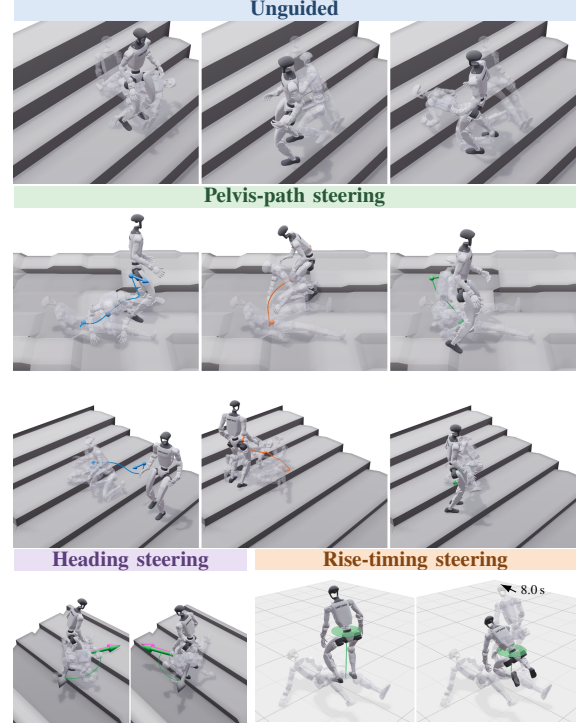


Fig. 7. **Test-time steering shapes recovery.** Rows show unguided recoveries, two pelvis-path steering examples, and heading (left) and rise-timing steering (right). Rise-timing panels show both motions at 6.5 s; the faint upright pose shows the later recovery at 8.0 s.

D. Test-Time Steering of Recovery

We next use the predicted motion intent to steer recovery toward task objectives: where the robot faces, when it rises, and which route it follows, using the costs of Sec. III-D. We use starts on flat ground, stairs, and a slope with a riser (Fig. 7) and two targets per objective, giving 18 commands with 100 trials each.

One trained model follows different objectives. Changing the target changes the executed heading, rise schedule, and route accordingly (Fig. 8), without retraining for each objective. Across the three terrains, steered recovery still succeeds in 95.4% of the 1,800 trials, with heading errors below 7° , timing errors below 0.7 s, and pelvis-path errors of 12–15 cm (Table IV). The predicted motion intent thus provides a single interface for choosing how recovery unfolds.

Terrain limits how freely recovery can be steered. Command following is hardest where the terrain constrains support. Heading errors are larger on stairs and the slope than on flat ground, and the slope has the largest timing errors.

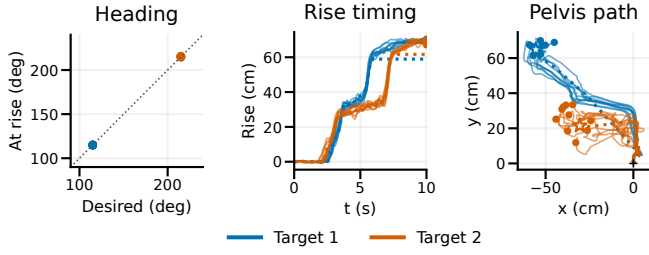


Fig. 8. **Executions follow their targets.** Ten successful flat-ground trials per target, with two targets from the same start in each panel. Heading is measured at the confirmed rise. Dotted lines and stars show the commands.

TABLE IV

ONE MODEL FOLLOWS HEADING, RISE-TIMING, AND PATH COMMANDS

Flat			Stairs			Slope + riser		
Target	Error	SR	Target	Error	SR	Target	Error	SR
<i>Heading steering: target and MAE (deg)</i>								
115	3.5	99	345	4.7	96	275	4.4	100
215	3.3	99	285	6.6	98	0	5.3	98
<i>Rise-timing steering: target rise time and MAE (s)</i>								
6.06	0.28	97	3.52	0.35	90	4.80	0.68	99
7.58	0.09	99	4.24	0.03	100	5.66	0.58	100
<i>Pelvis-path steering: path and RMSE (cm)</i>								
P1	12.5	97	P1	12.4	94	P1	12.2	89
P2	14.1	93	P2	12.0	80	P2	14.5	90

SR in %, over 100 trials per command. Each target pair shares a start. Targets and guidance step sizes are fixed before evaluation.

Path steering has the lowest success on these terrains (80–94% SR), and stairs account for half of all failures (Table IV). On these terrains, the commanded motion competes with the support the robot needs to stand up.

E. Ablation: Depth Perception and Prediction Order

We remove depth from the planner alone (*blind planner*) and from both modules (*both blind*). Each variant runs 120 unguided trials on uneven terrain from the same starts.

Depth in both modules supports controlled recovery. Removing planner depth lowers SR from 93.3% to 85.0%, even though the decoder still observes the terrain; removing decoder depth as well lowers it further to 79.2% (Table V). The blind variants also drift farther from the start, with mean pelvis displacement rising from 0.78 to 0.92 and 1.03 m, and peak torque and joint-limit excess follow the same order. Depth thus matters in both modules, most in the planner.

To test prediction order, we compare Ours with *joint prediction*, which denoises states and motion latents together and uses the same decoder and training budget. We steer both models toward heading, rise-timing, and pelvis-path targets with replanning intervals of 0.2, 0.5, and 1.0 s, using the same starts, targets, and guidance strengths.

Predicting states first makes steering more precise. At the default 0.2 s replanning interval, Ours lowers heading error from 7.1 to 5.3°, height-profile error from 3.0 to 1.8 cm, and pelvis-path error from 23.1 to 16.3 cm (Table VI). With less frequent replanning, errors grow and the gap narrows. Steering the predicted states before generating motion latents

TABLE V

DEPTH IN THE PLANNER AND DECODER IMPROVES RECOVERY

Variant	Depth		SR (%)	Pelvis displ. (m)	Peak torque (N m)	Limit excess (mrad)
	Planner	Decoder				
Ours	✓	✓	93.3	0.78	89.3	2.35
Blind planner	×	✓	85.0	0.92	93.6	3.38
Both blind	×	×	79.2	1.03	96.0	3.70

120 unguided trials per variant across uneven terrains.

TABLE VI

PREDICTING STATES FIRST LOWERS STEERING ERROR

Replan (s)	Variant	Heading		Rise timing		Pelvis path	
		SR (%)	Error (deg)	SR (%)	Error (cm)	SR (%)	Error (cm)
0.2	Ours	90.0	5.3	94.2	1.8	89.2	16.3
	Joint	89.2	7.1	93.3	3.0	78.3	23.1
0.5	Ours	96.7	7.3	94.2	3.0	72.5	23.3
	Joint	95.8	8.1	95.0	4.8	64.2	27.4
1.0	Ours	99.2	13.5	99.2	6.1	85.8	26.1
	Joint	76.7	13.8	94.2	6.0	83.3	29.2

Joint: joint prediction. 120 trials per objective, variant, and interval across terrains. Rise-timing error is height-profile RMSE.

conditions execution on the steered intent, whereas jointly denoised latents need not follow the steered states.

Joint prediction is less stable under steering. Under path steering at 0.2 and 0.5 s, joint prediction recovers in 78.3% and 64.2% of trials, vs. 89.2% and 72.5% for Ours. Under heading steering at 1.0 s, it recovers in 76.7% vs. 99.2%. When its latents drift from the steered states, execution can leave the plan and fail to recover. Predicting states first thus makes steering both more precise and more stable.

F. Real-World Experiments

Experimental setup. We test recovery diversity, steering, and terrain awareness on a 29-DoF Unitree G1 with proprioception and an egocentric depth camera. Inference runs offboard on an RTX 4090, with asynchronous planning and a 50 Hz decoder. DLIO [30] estimates pelvis motion from LiDAR and IMU measurements.

The prior’s alternatives carry over to hardware. Unguided recovery succeeds in every flat-ground trial from both prone and supine starts (Table VII), yet it does not repeat a single motion. Restoring the same commanded joint pose and resampling diffusion noise produces different final headings and positions (Fig. 10). The recorded sequences in Fig. 9D–F show supine recoveries that turn left, rise in place, and turn right from the same posture.

Steering selects among these alternatives. We command three target directions from each of the prone and supine poses on flat ground, and a recovery route in five trials beside a wall. All 11 steered trials recover (Table VII), following the requested direction or route across platforms, on flat ground, and beside a wall (Fig. 9A–C).

Depth adapts recovery to a wall, and steering sets its

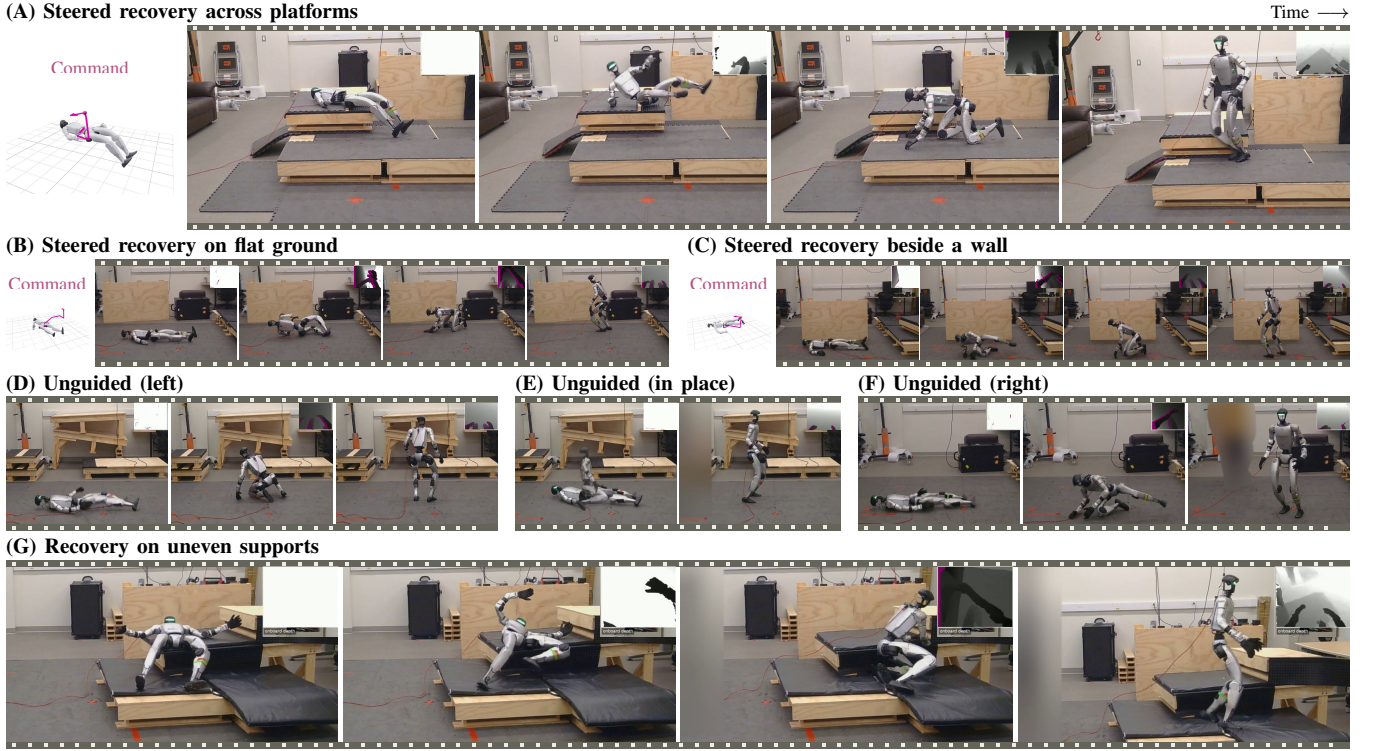


Fig. 9. **Real-world recovery diversity, steering, and terrain interaction.** (A–C) Steered recovery across platforms, on flat ground, and beside a wall; (D–F) unguided supine recoveries turning left, rising in place, and turning right; (G) recovery on uneven supports. Sequences progress from left to right; (E) includes a translucent intermediate pose. Insets show onboard depth; the view to the left of each steered sequence shows the commanded route in magenta.

TABLE VII
HARDWARE RECOVERY SUCCEEDS IN 40 OF 41 TRIALS

Unguided recovery		Steered recovery	
Setup	Success	Setup	Success
Flat, prone	10/10	Flat, prone	3/3
Flat, supine	10/10	Flat, supine	3/3
Stairs	9/10	Near wall	5/5

Successful recoveries to standing / trials.

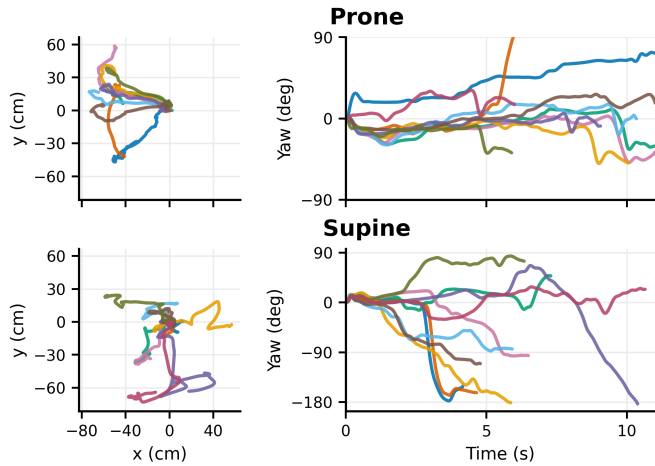


Fig. 10. **Repeated hardware recoveries vary in heading and position.** Pelvis motion of all flat-ground unguided runs from the same commanded prone or supine pose, relative to the start: x forward, y left, and continuous yaw from 0° . Colors distinguish runs.

direction. The wall example separates the roles of perception

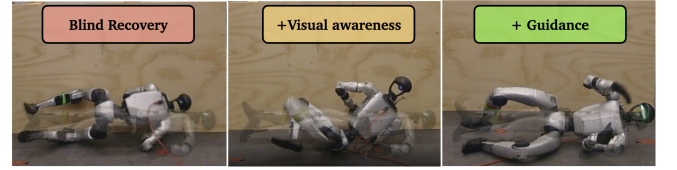


Fig. 11. **Vision and steering change recovery beside a wall.** Left to right: blind, depth-conditioned, and depth-conditioned steered trials. Overlaid poses show each recorded motion.

and steering (Fig. 11). From the same nominal pose, blind recovery rotates into the wall and collides with it. With depth, the robot first turns toward the wall and then changes direction toward the open side. With steering, it heads to the open side from the start. Depth thus lets recovery react to the obstacle, while steering expresses the preferred direction from the outset.

Recovery adapts to stairs. Unguided recovery succeeds in 9 of 10 stair trials, despite varying step geometry and initial body configuration (Table VII). The robot adjusts hand and foot support as it rises between levels (Fig. 9G), showing that the prior accommodates uneven contacts.

V. CONCLUSION AND FUTURE WORK

We present perceptive humanoid recovery that combines a multimodal recovery repertoire with test-time steering. Our visual planner predicts and steers motion intent before generating motion latents for feedback execution. In simulation, our approach recovers more reliably and with lower contact forces than prior controllers while keeping several routes

available. One trained model follows heading, rise-timing, and route objectives, and predicting states first makes this steering more precise and more stable. On a real humanoid, it recovers in nearly all trials and follows commanded directions and routes. Steering remains harder where terrain constrains support, and predicting recovery during dynamic interactions remains challenging. Future work will extend planning to changing scenes, where moving objects and external disturbances alter contact and support.

REFERENCES

- [1] 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,” in *Proceedings of Robotics: Science and Systems*, (LosAngeles, CA, USA), June 2025.
- [2] X. He, R. Dong, Z. Chen, and S. Gupta, “Learning Getting-Up Policies for Real-World Humanoid Robots,” in *Proceedings of Robotics: Science and Systems*, (LosAngeles, CA, USA), June 2025.
- [3] Z. Xu, Y. Li, K.-y. Lin, and S. X. Yu, “Unified humanoid fall-safety policy from a few demonstrations,” *arXiv preprint arXiv:2511.07407*, 2025.
- [4] O. Azulay, Z. Xu, A. Scheffer, and S. X. Yu, “VIGOR: Visual goal-in-context inference for unified humanoid fall safety,” *arXiv preprint arXiv:2602.16511*, 2026.
- [5] K. Zhao, G. Li, and S. Tang, “DartControl: A diffusion-based autoregressive motion model for real-time text-driven motion control,” in *International Conference on Learning Representations*, vol. 2025, pp. 23569–23592, 2025.
- [6] Q. Liao, T. E. Truong, X. Huang, Y. Gao, G. Tevet, K. Sreenath, and C. K. Liu, “BeyondMimic: From motion tracking to versatile humanoid control via guided diffusion,” *Science Robotics*, vol. 11, no. 117, p. eadx8924, 2026.
- [7] L. Kovar, M. Gleicher, and F. Pighin, “Motion graphs,” *ACM Trans. Graph.*, vol. 21, no. 3, pp. 473–482, 2002.
- [8] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, “Planning with diffusion for flexible behavior synthesis,” in *Proceedings of the 39th International Conference on Machine Learning*, vol. 162 of *Proceedings of Machine Learning Research*, pp. 9902–9915, PMLR, 2022.
- [9] 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 *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2521–2526, IEEE, 2002.
- [10] J. Stückler, J. Schwenk, and S. Behnke, “Getting back on two feet: Reliable standing-up routines for a humanoid robot,” in *International Conference on Intelligent Autonomous Systems*, pp. 676–685, 2006.
- [11] V. C. V. Kumar, S. Ha, and C. K. Liu, “Learning a unified control policy for safe falling,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3940–3947, IEEE, 2017.
- [12] C. Gaspard, M. Duclausaud, G. Passault, M. Daniel, and O. Ly, “FRASA: An end-to-end reinforcement learning agent for fall recovery and stand up of humanoid robots,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15994–16000, IEEE, 2025.
- [13] A. d’Avella and E. Bizzi, “Shared and specific muscle synergies in natural motor behaviors,” *Proceedings of the National Academy of Sciences*, vol. 102, no. 8, pp. 3076–3081, 2005.
- [14] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, “DeepMimic: Example-guided deep reinforcement learning of physics-based character skills,” *ACM Trans. Graph.*, vol. 37, no. 4, pp. 143:1–143:14, 2018.
- [15] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang, Z. Luo, G. He, N. Sobanbabu, C. Pan, Z. Yi, G. Qu, K. Kitani, J. K. Hodgins, L. Fan, Y. Zhu, C. Liu, and G. Shi, “ASAP: Aligning simulation and real-world physics for learning agile humanoid whole-body skills,” in *Proceedings of Robotics: Science and Systems*, 2025.
- [16] Z. Luo, Y. Yuan, T. Wang, C. Li, F. Castañeda, S. Chen, Z.-A. Cao, J. Li, D. Minor, Q. Ben, *et al.*, “SONIC: Supersizing motion tracking for natural humanoid whole-body control,” *Science Robotics*, vol. 11, no. 117, p. ead4592, 2026.
- [17] Y. Li, Z. Luo, T. Zhang, C. Dai, A. Kanervisto, A. Tirinzoni, H. Weng, K. Kitani, M. Guzek, A. Touati, *et al.*, “BFM-Zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning,” in *International Conference on Learning Representations*, vol. 2026, pp. 79697–79725, 2026.
- [18] T. Xie, H. Zhang, J. Park, Z. Wang, B. Wen, J. Li, X. Li, Q. Ben, H. Weng, Y. Ye, D. Minor, T. Wang, C. Jiang, S. Fidler, J. Kautz, L. Fan, Y. Zhu, Z. Luo, U. Iqbal, and Y. Yuan, “GRAIL: Generating humanoid loco-manipulation from 3D assets and video priors,” *arXiv preprint arXiv:2606.05160*, 2026.
- [19] C. Tessler, Y. Guo, O. Nabati, G. Chechik, and X. B. Peng, “Masked-Mimic: Unified physics-based character control through masked motion inpainting,” *ACM Trans. Graph.*, vol. 43, no. 6, pp. 209:1–209:21, 2024.
- [20] Z. Zhang, K. Wen, M. Xu, J. He, C. Li, T. Miki, C. Schwarke, C. Zhang, X. B. Peng, and M. Hutter, “Learning whole-body humanoid locomotion via motion generation and motion tracking,” *arXiv preprint arXiv:2604.17335*, 2026.
- [21] A. Ajay, Y. Du, A. Gupta, J. Tenenbaum, T. Jaakkola, and P. Agrawal, “Is conditional generative modeling all you need for decision-making?,” in *International Conference on Learning Representations*, 2023.
- [22] X. Huang, T. Truong, Y. Zhang, F. Yu, J. P. Sleiman, J. Hodgins, K. Sreenath, and F. Farshidian, “Diffuse-CLoC: Guided diffusion for physics-based character look-ahead control,” *ACM Trans. Graph.*, vol. 44, no. 4, 2025.
- [23] F. G. Harvey, M. Yurick, D. Nowrouzezahrai, and C. Pal, “Robust motion in-betweening,” *ACM Trans. Graph.*, vol. 39, no. 4, 2020.
- [24] Bones Studio, “BONES-SEED: Skeletal everyday embodiment dataset,” Motion Data by Bones Studio, 2026.
- [25] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, “AMASS: Archive of motion capture as surface shapes,” in *IEEE/CVF International Conference on Computer Vision*, 2019.
- [26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.
- [27] S. Ross, G. Gordon, and D. Bagnell, “A reduction of imitation learning and structured prediction to no-regret online learning,” in *International Conference on Artificial Intelligence and Statistics*, pp. 627–635, PMLR, 2011.
- [28] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Advances in Neural Information Processing Systems*, 2020.
- [29] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” in *International Conference on Learning Representations*, 2021.
- [30] K. Chen, R. Nemiroff, and B. T. Lopez, “Direct LiDAR-inertial odometry: Lightweight LIO with continuous-time motion correction,” in *IEEE International Conference on Robotics and Automation*, pp. 3983–3989, 2023.