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

Robot Crash Course: Learning Soft and Stylized Falling

Pascal Strauch, David Müller, Sammy Christen, Agon Serifi,
Ruben Grandia, Espen Knoop, Moritz Bächer

Fig. 1: We propose a reinforcement learning technique that balances user-guided stylized pose objectives and damage-minimizing soft falling objectives for bipedal and other legged robots.

***Abstract*— Despite recent advances in robust locomotion, bipedal robots operating in the real world remain at risk of falling. While most research focuses on preventing such events, we instead concentrate on the phenomenon of falling itself. Specifically, we aim to reduce physical damage to the robot while providing users with control over the robot's end pose. To this end, we propose a robot-agnostic reward function that balances the achievement of a desired end pose with impact minimization and the protection of critical robot parts during reinforcement learning. To make the policy robust to a broad range of initial falling conditions, and to enable the specification of an arbitrary and unseen end pose at inference time, we introduce a simulation-based sampling strategy of initial and end poses. Through simulated and real-world experiments, our work demonstrates that even bipedal robots can perform controlled, soft falls.**

I. INTRODUCTION

During dynamic motions, legged robots often encounter underactuated contact states that demand continuous dynamic balancing [1]. For bipedal robots, this issue is particularly pronounced, as they must control a heavy body on a relatively small area of support. Recent reinforcement learning–based locomotion controllers have made impressive progress in robustness [2], [3], yet the risk of falling in unstructured real-world environments remains substantial. As robots are pushed closer to their limits, much like for humans, certain disturbances or conditions will inevitably cause them to fall. However, unlike humans, robots usually fall in an uncoordinated and uncontrolled manner, leaving delicate components unprotected and breaking the illusion of lifelike motion.

A common approach is to improve controller robustness by adding domain randomization to policy training [4], integrating safety-oriented terms into optimization [5] or reward functions [6], or restricting the uncertainty by reducing the range of capabilities. While such approaches improve stability, they do not guarantee fall prevention in practice and may severely limit a robot's performance or capabilities. Rather than preventing a fall at all costs, we believe it is advantageous to embrace the potential of a fall, providing user control of end poses for stylization and ease of recovery.

In this paper, we therefore explore whether the robot can execute a fall in a controlled and visually-appealing manner. Falling is a challenging problem, as it requires performing contact-rich maneuvers within a very short time window and from a wide range of initial states. Moreover, for falling, multiple competing objectives need to be balanced, such as reducing impact, protecting critical components, and achieving desired motion characteristics.

Existing research on robot falling mostly focuses on a single objective or a controlled scenario. Once a fall is detected, common strategies are to *freeze* the actuators with high gains [7] or achieve a compliant reaction using low gains. Both approaches offer limited controllability over the resulting motion and suffer from high impact. More involved solutions often rely on hand-crafted fall strategies, such as executing predefined falling motions [8] or tracking predefined contact sequences [9]. This idea has recently been broadened to adaptive contact sequences, but remains restricted to a single falling direction [10], [11] or requires manual considerations tailored to specific fall scenarios, such as falling forwards or backwards [12].

In contrast, our method reduces overall impact forces and provides fine-grained user control, through the specification of critical components to protect and desired end poses for the robot to reach. This can be used for artistic control, as shown here, but could also serve as a starting pose for a recovery policy. We propose a reinforcement learning (RL) solution that offers adjustable trade-offs between damage reduction and pose objectives. To generalize across a wide range of user-specified end poses, we propose a physics-informed sampling strategy that comprehensively covers the

All authors are affiliated with Disney Research, Zurich, Switzerland
first.last@disneyresearch.com

distribution of initial and final states. Importantly, by leveraging reinforcement learning, our approach supports a wide variety of falling scenarios.

In our experiments, we compare our method quantitatively to standard falling strategies, showing that our approach results in softer falls. Through an ablation study in simulation, we demonstrate how our proposed method leads to controlled falling while adhering to landing in desired poses, with a user-defined trade-off between the two. In real-world experiments, we qualitatively demonstrate that our policies lead to falls without damage. To the best of our knowledge, this is the first general approach that demonstrates user-controlled falling of a bipedal robot in the real world. While we focus our evaluation on bipedal systems due to their inherent instability, our modeling is agnostic to the number of legs.

Succinctly, we contribute:

- A learning-based technique that balances impact minimization with a user-defined end pose, providing artistic control over a fall and facilitating a successful recovery.
- A sampling strategy of initial and end poses, enabling the training of a general falling policy that allows a user to specify an unseen and desired end pose at inference.
- Extensive ablations of our method in simulation and qualitative evaluation on a bipedal robot, highlighting the utility of our approach.

II. RELATED WORK

A. Soft Falling

Initial works on bipedal falling rely on hand-crafted strategies and predefined motions. A common approach is to treat falling as a controlled event that is managed with a sequence of carefully designed actions. For example, controllers can trigger specific joint trajectories, such as bending the robot's knees and extending its arms to reduce impact forces [12], [13], or tracking UKEMI-inspired falling motions [8], [9]. Alternatively, the robot can be guided by a predefined contact sequence [14]. As a complementary strategy, the gains of the actuators can be softened, making the joints more compliant to passively absorb impacts [12], [15].

A main limitation of these earlier works is their focus on relatively slow, walking robots and falls occurring primarily in the sagittal plane. As robots become capable of more dynamic motions [16], [17], the likelihood of multi-directional falls with high impact forces increases.

Recent advancements in RL allow for more general, flexible, and robust methods that require fewer assumptions. By allowing for adaptive and learned contact sequences, various individual fall strategies can be unified [10], [11], and scale beyond the simplification of sagittal plane falls [18]. There has been more recent focus on quadruped falling policies [19], [20]. ALMA [20], for example, provides a general framework that assigns time-varying damage-reduction rewards across the different phases of a fall.

We extend upon these related works by leveraging the strength of RL, and propose a general learning framework for soft falling that covers diverse falling scenarios. Our method accounts for sensitive robot parts, enabling the policy to minimize impacts on critical components, without manually specifying falling motions or contact sequences.

B. Stylized Falling

For applications in human-robot interaction, lifelike and stylistic robot motions become important [4], [21], [22]. In character animation, motion is typically defined by keyframes, with intermediate poses generated using various interpolation methods. Those methods range from simple parametric curves [23] to sophisticated learning-based interpolation techniques [24]. Keyframes were also explored in RL to sparsely define a robot's motion [25]. However, so far, artistic keyframes have only been applied in controlled settings, and not in the context of a falling objective. In fact, most works in simulated character control aim to prevent falling at all costs, either by introducing non-physical fictitious forces [26], employing early termination techniques [27], or explicitly penalizing falling through a reward [16].

We extend the capabilities of a falling policy by combining soft falling with style guidance, an aspect particularly relevant for human-oriented applications. Specifically, our framework enables steering the fall towards stylistic end poses while reducing the impact forces caused by it.

III. OVERVIEW

Falling motions are challenging as they arise from diverse and unstable initial states. Our goal is to enable a robot to perform a controlled fall across such conditions, while minimizing impact and reaching a user-specified, stylized end pose. Concretely, a user specifies two inputs: the relative sensitivity between robot components that are fixed at training time, and a desired end pose that is specified at inference time and should be reached by the robot once at rest, irrespective of the initial falling condition.

End poses can serve different goals. For instance, artists can define expressive poses to enable falls with an intended stylistic effect, turning a perceived failure into a believable motion. Moreover, end poses can be chosen to serve as suitable starting poses for recovery policies, facilitating a seamless transition into a standing pose [28], [29].

Note that this work focuses exclusively on a graceful falling behavior of a robot, and not on deciding whether a fall should occur.

IV. METHOD

To achieve our goals as outlined above, our method trains a policy via reinforcement learning, relying on a reward function that balances impact minimization with pose objectives (refer to Fig. 2). Before describing our rewards, sampling strategy, and initialization procedure, we introduce the domain-specific states, goals, and actions.

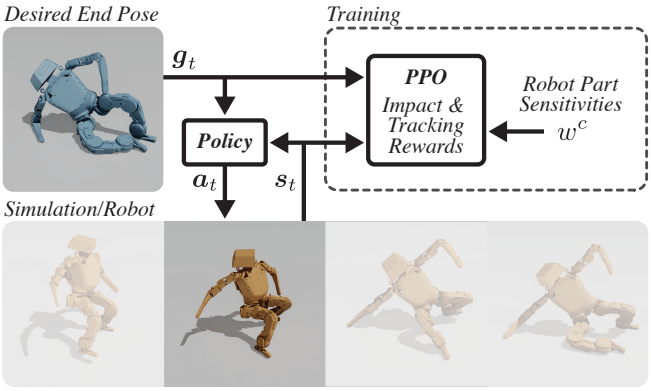

Fig. 2: **Method Overview.** We leverage reinforcement learning to train a robust falling policy (right). Our method learns to balance **impact minimization** with reaching a **desired end pose** through our reward formulation, which considers user-specified robot part sensitivities. During inference (left), the policy is guided by a user-specified end pose, while simultaneously minimizing impact.

A. Reinforcement Learning Setup

Our goal is to determine a sequence of actions $\boldsymbol{a}_t$ through a policy $\pi(\boldsymbol{a}_t|\boldsymbol{s}_t, \boldsymbol{g}_t)$ that transitions the robot from an initial state $\boldsymbol{s}_0$ into a final state $\boldsymbol{s}_T$, taking on a user-specified end pose, with $t \in [0, T]$. The actions $\boldsymbol{a}_t$ are joint position setpoints for proportional-derivative (PD) controllers, and the proprioceptive state is

$$\boldsymbol{s}_t := \big(\boldsymbol{\theta}_t, \boldsymbol{v}_t, \boldsymbol{\omega}_t, \boldsymbol{q}_t, \dot{\boldsymbol{q}}_t, \boldsymbol{a}_{t-1}, \boldsymbol{a}_{t-2}\big), \tag{1}$$

where $\boldsymbol{\theta}_t$ is the root orientation, represented with a unit quaternion, $\boldsymbol{v}_t$ and $\boldsymbol{\omega}_t$ are the root's linear and angular velocities, and $\boldsymbol{q}_t$ and $\dot{\boldsymbol{q}}_t$ are its joint angles and joint angular velocities.

The time-varying goal

$$\boldsymbol{g}_t := \big(\hat{\boldsymbol{\theta}}_t, \hat{\boldsymbol{q}}_t\big) \tag{2}$$

is derived from the user-specified end pose $\boldsymbol{g} = \big(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{q}}\big)$ as outlined below, where $\hat{\boldsymbol{\theta}}_t$ is the robot's target root orientation and $\hat{\boldsymbol{q}}_t$ its target joint configuration.

To make our policy invariant under the robot's global pose, we represent the root orientation in states and goals in the local path frame, and the root's linear and angular velocities w.r.t. the root frame. The path frame is defined with the root at the origin, x- and y-axes in the horizontal plane, and the x-axis aligned with the root's facing direction.

Note that we do not vary the end goal but apply the time-varying transformation from the end pose to path-relative coordinates.

During training, we sample initial states $\boldsymbol{s}_0$ and end poses $\boldsymbol{g}$ as described in Secs. IV-C and IV-D.

B. Reward Design

The reward function is designed to balance accurate *end pose tracking* with *soft impact*, supplemented by *regularization* terms.

TABLE I: **Weighted Reward Terms.** To penalize contact forces, we sum up all contact forces that act on a component c in the force vectors $\boldsymbol{f}_t^c$, then multiply them with the component's sensitivity weight w^c. $\dot{\boldsymbol{v}}_t$ is the linear acceleration of the root, and $\boldsymbol{\tau}_t$ and $\ddot{\boldsymbol{q}}_t$ are joint torques and accelerations. For end pose tracking of the root orientation, we apply Rodrigues' rotation formula to convert unit quaternions to rotation matrices, measuring differences in yaw rotations between the simulated and the goal state ($\boldsymbol{e}_z$ is the unit vector along the z-axis).

Name	Reward Term	Weight
	Impact Reward	
Contact forces	$-\sum_{\text{comp. } c} \|w^c \boldsymbol{f}_t^c\|_\infty^2$	200
Root acc.	$-\|\dot{\boldsymbol{v}}_t\|_2^2$	0.2
	Tracking Reward	
Root orientation	$-u(t)\|\boldsymbol{R}(\boldsymbol{\theta}_t)^T \boldsymbol{e}_z - \boldsymbol{R}(\hat{\boldsymbol{\theta}}_t)^T \boldsymbol{e}_z\|_2^2$	20.0
Joint positions	$-u(t)\|\boldsymbol{q}_t - \hat{\boldsymbol{q}}_t\|_2^2$	1.0
	Regularization Reward	
Joint torques	$-\|\boldsymbol{\tau}_t\|_2^2$	$1.0 \cdot 10^{-3}$
Joint acc.	$-\|\ddot{\boldsymbol{q}}_t\|_2^2$	$7.5 \cdot 10^{-7}$
Action rate	$-\|\boldsymbol{a}_t - \boldsymbol{a}_{t-1}\|_2^2$	0.1
Action acc.	$-\|\boldsymbol{a}_t - 2\boldsymbol{a}_{t-1} + \boldsymbol{a}_{t-2}\|_2^2$	0.05
Positive offset	1.0	50

$$r_t = r_t^{\text{tracking}} + r_t^{\text{impact}} + r_t^{\text{regularization}}. \tag{3}$$

A detailed breakdown of the weighted reward terms is provided in Tab. I, where hats $\hat{\cdot}$ denote target quantities derived from $\boldsymbol{g}$.

To promote soft impacts, we adopt an impact reward r_t^{impact} (Tab. I, *top*) inspired by prior work on quadrupedal falling [20]. We extend the contact force reward by scaling contact forces of robot components with non-negative sensitivity weights w^c. Root acceleration is also penalized to discourage abrupt motion, regardless of contact.

The end pose tracking reward r_t^{tracking} (Tab. I, *middle*) compares the simulated robot pose to a target end pose and incentivizes the policy to reach this pose. Specifically, the tracking reward combines a joint tracking and a global yaw-invariant orientation term. To encourage the policy to initially focus on reducing impact forces, before smoothly transitioning to pose tracking, the tracking reward is modulated with a time-dependent cubic spline $u(t)$, which interpolates the reward between 0.0 and 1.0 over the blending duration T_{blend}:

$$u(t) = \begin{cases} -2\left(\dfrac{t}{T_{\text{blend}}}\right)^3 + 3\left(\dfrac{t}{T_{\text{blend}}}\right)^2, & 0 \le t \le T_{\text{blend}}, \\ 1.0, & t > T_{\text{blend}}. \end{cases} \tag{4}$$

The parameter T_{blend} is empirically determined to balance impact reduction and tracking performance, and satisfies $T_{\text{blend}} \le T$. In practice, we find that extending learning

TABLE II: **Initial Robot State Ranges.** During training, we cover a wide range of initial falling conditions by sampling from the ranges below.

Variable	Range
Roll, Pitch [rad]	$[-\pi/6, \pi/6]$
Linear root velocity [$\mathrm{m\,s^{-1}}$]	$[-2.0, 2.0]$
Angular root velocity [$\mathrm{rad\,s^{-1}}$]	$[-0.5, 0.5]$
Joint velocities [$\mathrm{rad\,s^{-1}}$]	$[-0.5, 0.5]$

beyond the interpolation time helps the policy come to rest and maintain the final pose without jitter.

Following prior work [4], [27], we add regularization rewards $r_t^{\text{regularization}}$ (Tab. I, *bottom*) to penalize excessive joint torques and encourage smooth actions, helping to avoid vibrations and unnecessary effort. A constant positive reward ensures that the agent observes positive rewards from the start of training, which facilitates learning [30].

C. Sampling-Based End Pose Generation

To enable user control over a wide range of end poses at inference time, we introduce a physics-informed sampling mechanism to generate a dataset of statically stable and feasible robot configurations.

We begin by sampling random joint configurations within feasible limits and discarding all configurations that result in self-collision between robot components. Next, we sample the robot's root orientation by applying first pitch and then roll rotations around the robot's axis, both over the full range of $\pm 180°$.

To obtain statically stable end poses, each sampled configuration is initialized and dropped from a predefined height of $0.04\,\mathrm{m}$ with actuators frozen (high gains with fixed setpoint), until the robot comes to rest, as visualized in the supplemental video material.

This procedure, however, can produce a biased distribution: certain root orientations, such as poses on the robot's back, may be overrepresented, while others, like poses on the side, are underrepresented. To mitigate this bias, we iteratively sample new poses while discarding those that are already sufficiently represented, ensuring uniform coverage across root orientation bins.

We leverage Isaac Sim [31] to perform collision detection and to let frozen robots settle into statically-stable configurations, enabling GPU-accelerated generation of large batches of physically-feasible end poses.

D. Robot Initialization

To cover a wide range of possible initial falling states in which the policy may be activated, we randomize the initial conditions at the beginning of each episode. We sample a root orientation together with a joint configuration within feasible limits, and filter all poses to avoid ground penetration or self-collisions. Since our method is invariant to the global yaw angle, only the pitch and roll are varied, sampled in pitch–roll order. To further increase variability and to mimic the effects of external perturbations and unstable (i.e., falling) starting conditions, we also assign initial velocities to both the joints and the root. The corresponding parameter ranges are summarized in Tab. II.

TABLE III: **PPO Hyperparameters.** The hyperparameters used to train the falling policy.

Param.	Value
Num. iterations	75 000
Batch size (num. envs. $\times$ steps)	4096×24
Num. mini-batches	4
Num. epochs	5
Clip range	0.2
Entropy coefficient	0.0
Discount factor	0.99
GAE discount factor	0.95
Desired KL-divergence	0.01
Max gradient norm	1.0

TABLE IV: **Disturbance Forces.** We add random forces and torques to each specified body part, drawn from uniform distributions with the magnitudes listed below and applied per dimension. Disturbances are applied for a random "on" duration, followed by a random "off" duration before the next application.

Param.		Hips, Feet, Elbows	Pelvis, Head
Force [N]	XY	[0.0, 5.0]	[0.0, 5.0]
	Z	[0.0, 5.0]	[0.0, 5.0]
Torque [N m]	XY	[0.0, 0.25]	[0.0, 0.25]
	Z	[0.0, 0.25]	[0.0, 0.25]
Duration [s]	On	[0.25, 2.0]	[2.0, 10.0]
	Off	[1.0, 3.0]	[1.0, 3.0]

V. EXPERIMENTS AND RESULTS

We first outline implementation (Sec. V-A) and experimental details (Sec. V-B). Our evaluations are then organized into four parts. First, we compare our method against default falling strategies (Sec. V-C). Next, we perform extensive ablations of our approach that highlight the effectiveness of our reward formulation and the sampling-based end pose generation (Sec. V-D). We then showcase how the robot part sensitivity weights affect the resulting impact forces (Sec. V-E). Finally, we demonstrate the transferability of our approach from simulation to the real world through experiments with a bipedal robot (Sec. V-F).

A. Implementation Details

In the following, we provide details on the network architecture, training durations, and other relevant information to facilitate reproducibility.

We train the falling policy using PPO [32] within an asymmetric actor–critic setup [33], with hyperparameters listed in Tab. III and an adaptive learning rate [34]. To mitigate the sim-to-real gap, we add standard Gaussian noise to the inputs to the actor network and small disturbance forces listed in Tab. IV. In addition to the standard observations, the critic receives privileged observations including noiseless quantities, friction parameters, rigid body velocities and

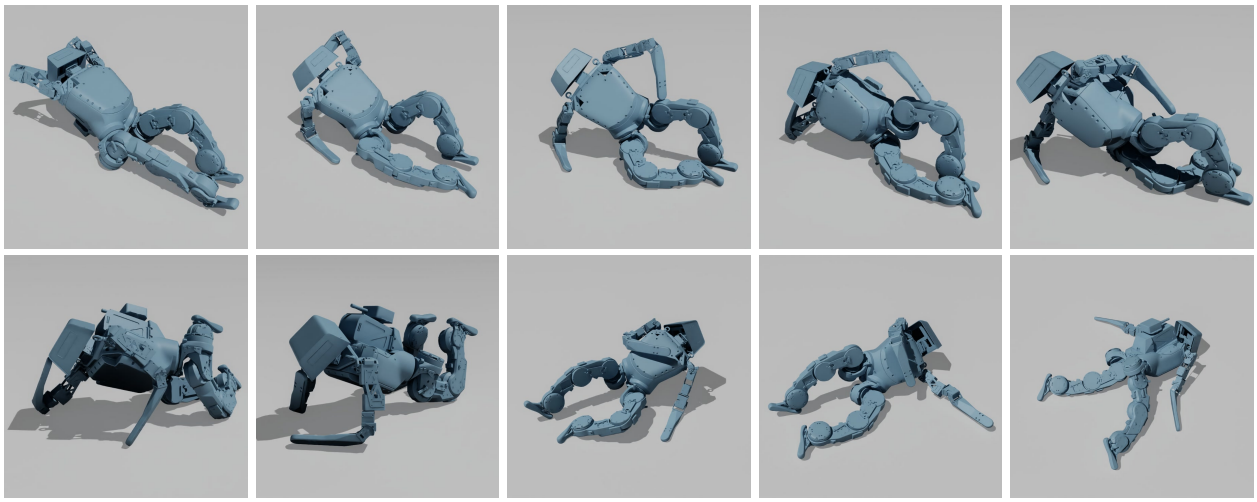

Fig. 3: **Artist-Designed End Poses.** Visual examples of the 10 artist-designed end poses used in our experiments.

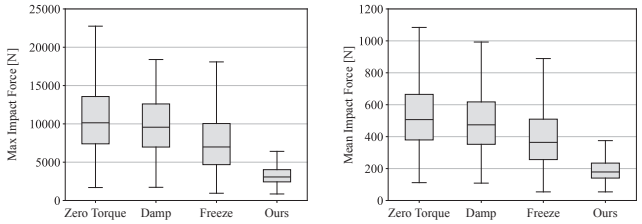

Fig. 4: **Impact Analysis.** Comparison of maximal (left) and mean (right) impact forces across body parts between standard falling strategies and our method.

accelerations, and the phase of the episode. Both the policy and value function are modeled using multi-layer perceptron (MLP) networks with ELU activations [35], comprising three layers with 512 units each. Observations are normalized using a running mean, following the standard practice in PPO [32]. Our simulations are performed using the GPU-accelerated Isaac Sim physics engine [31], running 4096 environment instances in parallel on a single RTX 4090 GPU. We train our falling policy for 75k iterations (approx. 48 h).

Next, we detail additional aspects of the reward formulation. When resolving contacts, the physics engine can generate excessively large forces, which is why, during training only, we clip values above 10 kN to improve numerical stability. To account for varying sensitivity across robot components, we assign different sensitivity weights. Previous experiments with our bipedal robot revealed that most damage occurred at the head, followed by the shoulders and elbows. To reflect this varying sensitivity, we assign the following sensitivity weights to the body parts: The pelvis and legs are weighted 1.0, elbows 2.0, shoulders 3.0, and head 4.0. In Sec. V-E, we perform an ablation of this weighting scheme.

B. Experimental Details

1) Data: Our training dataset comprises 24k target end poses, complemented by a test set of 2k poses, generated through our sampling-based generation (Sec. IV-C). Additionally, we employ a set of 10 expressive, artist-designed poses in various orientations to explore the generality of our method. These poses were manually created by artists in Blender [36], respecting the joint limits and avoiding self-penetration during the process, but ignoring physical constraints. The end poses shown in Fig. 3 are visual examples of the artist-designed poses in simulation. In Sec. V-D.3, we analyze the effect of dataset size on policy training.

2) Metrics: We base our evaluation on [18], [20] by comparing damage criteria across different control strategies. Unless specified otherwise, we evaluate the metrics over 32,768 trials with randomly sampled initial states and unseen target end poses from the test set. We use the following metrics throughout our experiments:

- **Max Impact Force:** The maximum impact force experienced by the robot during each rollout and across body parts.
- **Mean Impact Force:** The maximum of the mean impact force over all body parts experienced by the robot during each rollout.
- **Mean Root Orientation Error (MROE):** The mean root orientation error is given as the geodesic distance between the global yaw axis-aligned target end pose orientation and the robot's root orientation at the final time step of an episode.
- **Mean Joint Tracking Error (MJE):** The mean absolute joint tracking error over all joints at the last timestep of an episode.

3) Robot: We run experiments on a custom-built bipedal robot with 20 degrees of freedom (DoF), a total mass of 16.2 kg and a height of 0.84 m. Each leg has 5 DoF with Unitree A1 actuators, and the arms and neck are equipped with Dynamixel XH540-V150-R actuators. Our policy predicts actuator positions at 50 Hz that are passed to proportional-derivative (PD) controllers at each joint. We estimate the robot's state by fusing information from an onboard inertial measurement unit and motion capture.

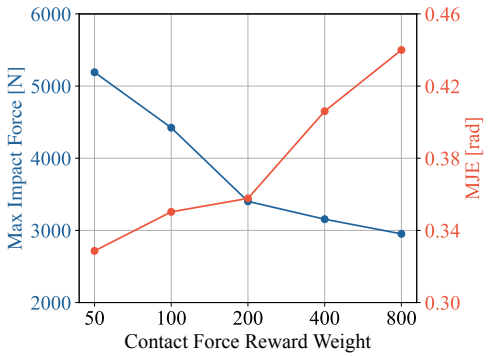

Fig. 5: **Impact vs. Tracking Ablation.** We measure the max impact force and mean joint tracking error for varying impact reward weights. Displayed are the mean values over all trials.

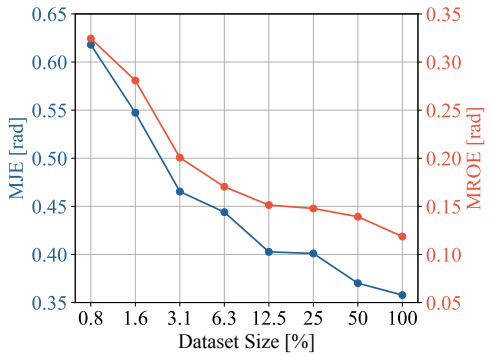

Fig. 6: **Dataset Size.** Mean joint tracking error and mean root orientation error for varying dataset sizes.

TABLE V: **Sampling-Based End Pose Ablation**. We compare two variants of our method, once trained with our sampling-based end pose generation (*generated*) and once trained on artist-designed poses (*artistic*). We report the mean and standard deviation of the mean joint position tracking error (MJE) and the mean root orientation error (MROE).

Test Data	**Training Data**	**MJE** [rad]	**MROE** [rad]
Generated	Generated	0.36 ± 0.10	0.12 ± 0.12
Generated	Artistic	1.03 ± 0.20	1.05 ± 0.58
Artistic	Generated	0.30 ± 0.09	0.09 ± 0.07
Artistic	Artistic (seen)	0.17 ± 0.12	0.08 ± 0.15

TABLE VI: **Impact Reduction of Critical Components.** Comparison of the baseline and the policy trained with increased sensitivity weight on the battery. We report the mean joint tracking error (MJE) and mean root orientation error (MROE) as mean with standard deviation, and battery impact forces as median and 95th percentile, reflecting their highly non-symmetric statistics.

Policy	**MJE** [rad]	**MROE** [rad]	**Median/95th %** [N]
w/o battery	0.32 ± 0.10	0.11 ± 0.11	36.12/3321.75
w/ battery	0.42 ± 0.11	0.16 ± 0.14	0.00/810.69

C. Comparison with Standard Falling Methods

Following the evaluations in [7], [11], [20], we compare our method against standard falling strategies commonly used in practice: applying zero torque, damping the actuators with low gains ($0.1\times$nominal), and freezing them with high gains ($10\times$nominal) at their most recent setpoints. As shown in Fig. 4, our method substantially reduces both the max and mean impact forces compared to the baselines and exhibits much lower variance. Furthermore, the falling dynamics with our method are controlled and predictable. In contrast, freezing the joints makes the robot behave as a single rigid body, falling in the initial direction, while damping or zero-torque settings produce interactions between components, resulting in more complex and less predictable motion. Please refer to our supplemental video for visual evidence of these insights. Overall, these results highlight the benefits of our approach over common existing falling strategies.

D. Ablations

1) Impact vs. Tracking Ablation: We ablate policies trained with varying weights of the contact force reward (see Tab. I) and evaluate the resulting maximal impact forces and mean joint tracking error. We illustrate the results in Fig. 5. As expected, increasing the contact force weight reduces impact forces but increases the joint tracking error. This highlights the inherent trade-off between minimizing impact and accurately reaching the target end pose. We found that a contact force weight of 200 provides a reasonable balance between these objectives.

2) Sampling-Based End Pose Generation: To evaluate our sampling-based end pose generation technique (see Sec. IV-C), we compare our approach trained on sampled end poses (*generated*) with a variant trained solely on artist-designed end poses (*artistic*). We report results in Tab. V. On the *generated* test set, the policy trained on generated end poses outperforms the variant trained on a few artist-designed end poses significantly in both mean joint tracking error and mean root orientation error.

On the other hand, our method trained on generated end poses achieves slightly higher mean root orientation and joint tracking errors on the artist-designed test set. Note, however, that the artist-trained variant has seen these poses during training. This leads to overfitting as indicated by its high errors on the generated test dataset. In contrast, our method generalizes well to unseen poses, even when drawn from a different data distribution (*artistic*).

3) Dataset Size: We examine how the number of generated end poses affects generalization to unseen end poses. We train multiple variants of our method, each using a progressively smaller subset of the full training dataset. We report the mean joint tracking error and mean root orientation error on our test set of unseen end poses, and illustrate the results in Fig. 6. We find that the best performance is achieved with our full dataset, yielding improved joint and

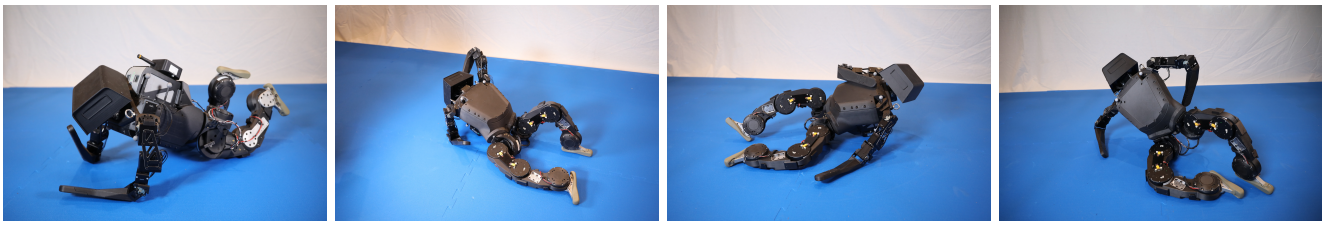

Fig. 7: **Real-World Experiments.** Qualitative examples of the artist-designed end poses the robot reached after falling.

orientation tracking. Dataset size is most critical in low-data regimes (1%–6% of the total dataset), indicating that a minimum amount of data is needed for generalization. Beyond this range, additional data continues to improve performance; however, the gains become more marginal.

E. Impact Reduction of Critical Components

Our method accounts for the varying sensitivity of different robot components. To evaluate this formulation, we split the pelvis into the main body and a rear battery pack, assigning a high sensitivity weight of 5.0 to the battery and 1.0 to all other components. This simulates a robot carrying sensitive hardware on its back. We compare a policy trained with these weights to a policy that has all sensitivity weights set to 1.0. The results in Tab. VI show a significant reduction in the 95th percentile, demonstrating that worst-case impacts can be greatly reduced. A median of 0.0 indicates that, in most falling scenarios, forces on the backpack can be fully eliminated. Thus, our method provides a general framework to balance tracking performance and impact forces on critical components. We provide qualitative results of this experiment in our supplemental video.

F. Real-World Experiments

We perform a set of qualitative real-world experiments to demonstrate the transferability of our method from simulation to the real world using the bipedal robot described in Sec. V-B). We select 10 artist-designed end poses and vary the initial conditions by randomly applying external forces to the robot with a stick. We then record the resulting falling behavior, with end poses illustrated in Fig. 7 and the entire falling motions shown in our supplemental video. Notably, we performed all of our experiments with a single robot, which remained fully functional throughout the experiments and showed no noticeable damage. This indicates that our method enables soft falling behavior that protects the robot's most sensitive part, regardless of the falling direction.

VI. DISCUSSION

Our approach shows promising results for bipedal falling, but has several limitations. Our experiments were all carried out with the same humanoid robot. While our modeling is agnostic to the robot morphology, future research could explore how well our method transfers to different humanoids or legged robots in general.

We study falling in an isolated manner and intentionally place the robot in unstable states that result in falling. A practical, real-world deployment of our approach would require a mechanism that predicts unstable states to trigger appropriate falling motions. To anticipate a fall, simple heuristics, such as detecting invalid state estimates, insufficient battery, or other safety-critical conditions, could be used. Future work could explore predicting a fall from the robot's motion dynamics. With such a fall detection system in place, our policy's transfer to real-world conditions could be more rigorously validated across a variety of falling scenarios.

In our approach, the impact weight per robot part must be defined prior to training. An exciting future avenue is the exploration of a policy that enables the adjustment of the policy's objectives at inference time, similar to multi-objective RL approaches [37]. This would allow users to, for example, increase the weight of impacts on components that are nearing their wear-and-tear limits.

Moreover, in our presented experiments, we pre-selected the target end poses. An interesting direction for future work is to automatically determine the most suitable falling pose based on the robot's initial state.

Finally, while we focused on stylized and soft falling, this behavior is tightly coupled with recovery, which has been explored in recent works [28], [29]. Future work could investigate how to best combine the training of falling and recovery policies, taking stylization into consideration in both policies.

VII. CONCLUSION

Falling remains an inevitable possibility for legged robots, and in this work, we have shown that a purpose-trained RL controller is able to both reduce the impact and severity of the fall, and also prescribe the pose in which the robot ends up. We have evaluated our method with both simulated and real-world experiments.

Falling means to temporarily relinquish control of the system. However, if the final state of the fall can be controlled, and damage can be mitigated, this also opens the potential for deliberately exploiting falls during robot operation. This could be applicable for stunt robots and slapstick performances, but could also be exploited in the future to traverse more extreme terrain.

ACKNOWLEDGMENT

We sincerely thank Violaine Fayolle and Doriano van Essen for designing the artistic poses used in this project.

REFERENCES

[1] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. D. Prete, "Optimization-based control for dynamic legged robots," *Trans. Rob.*, vol. 40, p. 43–63, Jan. 2024.

[2] Z. Zhuang, S. Yao, and H. Zhao, "Humanoid parkour learning," in *Conference on Robot Learning*. PMLR, 2025, pp. 1975–1991.

[3] H. Duan, B. Pandit, M. S. Gadde, B. Van Marum, J. Dao, C. Kim, and A. Fern, "Learning vision-based bipedal locomotion for challenging terrain," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 56–62.

[4] R. Grandia, E. Knoop, M. A. Hopkins, G. Wiedebach, J. Bishop, S. Pickles, D. Müller, and M. Bächer, "Design and Control of a Bipedal Robotic Character," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024.

[5] G. Romualdi, S. Dafarra, G. L'Erario, I. Sorrentino, S. Traversaro, and D. Pucci, "Online non-linear centroidal mpc for humanoid robot locomotion with step adjustment," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 412–10 419.

[6] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, vol. 9, no. 89, p. eadi9579, 2024.

[7] C. G. Atkeson, B. P. W. Babu, N. Banerjee, D. Berenson, C. P. Bove, X. Cui, M. DeDonato, R. Du, S. Feng, P. Franklin, M. Gennert, J. P. Graff, P. He, A. Jaeger, J. Kim, K. Knoedler, L. Li, C. Liu, X. Long, T. Padir, F. Polido, G. G. Tighe, and X. Xinjilefu, "No falls, no resets: Reliable humanoid behavior in the darpa robotics challenge," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 623–630.

[8] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2002, pp. 2521–2526.

[9] K. Ogata, K. Terada, and Y. Kuniyoshi, "Falling motion control for humanoid robots while walking," in *2007 IEEE-RAS 7th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2007, pp. 306–311.

[10] S. Ha and C. K. Liu, "Multiple contact planning for minimizing damage of humanoid falls," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2761–2767.

[11] V. C. 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)*, 2017, pp. 3940–3947.

[12] T. Ishida, Y. Kuroki, and T. Takahashi, "Analysis of motions of a small biped entertainment robot," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 1. IEEE, 2004, pp. 142–147.

[13] S. Wang and K. Hauser, "Real-time stabilization of a falling humanoid robot using hand contact: An optimal control approach," in *2017 IEEE-RAS 17th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2017, pp. 454–460.

[14] S. K. Yun and A. Goswami, "Tripod fall: Concept and experiments of a novel approach to humanoid robot fall damage reduction," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2799–2805.

[15] V. Samy and A. Kheddar, "Falls control using posture reshaping and active compliance," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 908–913.

[16] A. Serifi, R. Grandia, E. Knoop, M. Gross, and M. Bächer, "Vmp: Versatile motion priors for robustly tracking motion on physical characters," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '24. Goslar, DEU: Eurographics Association, 2024, p. 1–11.

[17] Q. Liao, T. E. Truong, X. Huang, G. Tevet, K. Sreenath, and C. K. Liu, "Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion," *arXiv:2508.08241*, 2025.

[18] V. Samy, K. Bouyarmane, and A. Kheddar, "Qp-based adaptive-gains compliance control in humanoid falls," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4762–4767.

[19] Y. Wang, M. Xu, G. Shi, and D. Zhao, "Guardians as you fall: Active mode transition for safe falling," in *2024 IEEE International Automated Vehicle Validation Conference (IAVVC)*. IEEE, 2024, pp. 1–8.

[20] Y. Ma, F. Farshidian, and M. Hutter, "Learning arm-assisted fall damage reduction and recovery for legged mobile manipulators," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 149–12 155.

[21] S. Christen, D. Müller, A. Serifi, R. Grandia, G. Wiedebach, M. A. Hopkins, E. Knoop, and M. Bächer, "Autonomous human-robot interaction via operator imitation," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025, pp. 18 724–18 731.

[22] A. F. Alvarez, F. Zargarbashi, H. Liu, S. Wang, L. Edwards, J. Anz, A. Xu, F. Shi, S. Coros, and D. W. Hong, "Learning to walk in costume: Adversarial motion priors for aesthetically constrained humanoids," in *2025 IEEE-RAS 24th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2025, pp. 593–600.

[23] J. Lee and S. Y. Shin, "A hierarchical approach to interactive motion editing for human-like figures," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '99. USA: ACM Press/Addison-Wesley Publishing Co., 1999, p. 39–48.

[24] D. Agrawal, J. Buhmann, D. Borer, R. W. Sumner, and M. Guay, "Skel-betweener: a neural motion rig for interactive motion authoring," *ACM Trans. Graph.*, vol. 43, no. 6, Nov. 2024.

[25] F. Zargarbashi, J. Cheng, D. Kang, R. W. Sumner, and S. Coros, "Robotkeyframing: Learning locomotion with high-level objectives via mixture of dense and sparse rewards," in *Conference on Robot Learning*, 2024.

[26] Y. Yuan and K. Kitani, "Residual force control for agile human behavior imitation and extended motion synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 763–21 774, 2020.

[27] 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, pp. 143:1–143:14, 2018.

[28] T. Tao, M. Wilson, R. Gou, and M. van de Panne, "Learning to get up," in *ACM SIGGRAPH 2022 Conference Proceedings*, ser. SIGGRAPH '22. New York, NY, USA: Association for Computing Machinery, 2022.

[29] 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.

[30] R. Sullivan, A. Kumar, S. Huang, J. Dickerson, and J. Suarez, "Reward scale robustness for proximal policy optimization via dreamerv3 tricks," *Advances in Neural Information Processing Systems*, vol. 36, pp. 1352–1362, 2023.

[31] NVIDIA, "Isaac Sim." [Online]. Available: https://github.com/isaac-sim/IsaacSim

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

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

[34] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 91–100.

[35] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[36] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[37] L. N. Alegre, A. Serifi, R. Grandia, D. Müller, E. Knoop, and M. Bächer, "Amor: Adaptive character control through multi-objective reinforcement learning," in *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, ser. SIGGRAPH Conference Papers '25. New York, NY, USA: Association for Computing Machinery, 2025.