Extended Abstract

Motivation Dealing with out-of-distribution situations is a major challenge in robotics in general, and is even more difficult for humanoid robots given their higher center of gravity and their movements' dissimilarity to human movement. The goal of this project is to explore two different methods of training for unexpected situations and/or changes to the robot mid-motion and comparing the achieved performance from these methods to a normal training method with no intentional configuration or environmental perturbances.

Method We employ curriculum learning and split our training paradigm into three stages, where the model is trained on progressively harder versions of the task. Throughout, we refer to the term "blinking", which we use to mean masking out certain inputs/outputs to the robot for a fixed duration of time, after which we restore them and randomly select another input/output to mask. We use PPO (Proximal Policy Optimization), which has an Actor-Critic architecture with two neural networks.

Stage 1: Training a baseline with no projectiles and no blinking.

Stage 2: Continuing training from stage 1, we now add projectiles into the training environment (still no blinking). These projectiles spawn at randomized positions on a cylinder centered around the robot's center of mass (CoM) and aim for the robot's CoM, which results in projectiles making contact with different parts of the robot with relatively uniform coverage.

Stage 3: Continuing training from stage 2, we now try four different types of blinking during training: no blinking (direct continuation of stage 2), proprioceptive (joint velocity) blinking, actuator limpness (joint motor) blinking, and exteroceptive (synthetic radar) blinking.

Implementation Stage 1, walking, was trained for 300 iterations and converged at around 150 iterations. Stage 2 was also trained for 100 iterations. In Stage 3, each blinking method was trained for 100 iterations. For each run, we used num_envs = 4096, meaning there were 4096 parallel simulated robots collecting experience at once.

Blinking was achieved by applying a mask that would zero out an element within the category (proprioceptive, actuator, or exteroceptive) at a set probability (default 0.01 for proprioceptive and exteroceptive, and 0.002 for actuator). This masking was held for a set amount of time (default 50 iterations for proprioceptive and exteroceptive, and default 30 iterations for actuator limpness) and then randomized again.

Results We found that blinking the actuator limpness of the humanoid was the most effective, with a mean survival time (episode length) of 1363.57 steps, while proprioceptive blinking only achieved 483.1 steps and exteroceptive blinking only achieved 524.84 steps. Blinking all three and training for 100 iterations yielded a mean survival time of 1232.87 steps.

Discussion These results imply that masking joint velocity or radar inputs failed to act as an effective regularizer. Rather than forcing the policy to develop more robust behavior, the network appeared to compensate through state estimation to infer the missing information, resulting in executing the same locomotion pattern as it would have without blinking. On the other hand, actuator blinking proved to be a powerful regularizer. This is likely due to how actuator masking physically alters the MDP of the training goal itself, forcing the policy to adapt.

Conclusion Our results suggest that actuator masking drives adaptability. Blinking actuator limpness prevents the actor neural network in PPO from over-relying on specific joints for stability, reducing catastrophic fall rates from 24% to 19% in hazardous environments. Future work could explore differently designed hazardous environments and blinking additional components of the humanoid, such as real camera feed.

BLIND: Bipedal Locomotion with Intermittent Navigation Data for Environmental Hazards

Team Members: Iris Xu, Eric Liang, Jamin Xie

Emails: izxu28@stanford.edu, ehliang@stanford.edu, jmx@stanford.edu

Abstract

Dealing with out-of-distribution situations is a major challenge in robotics, and is even more difficult for humanoid robots given their higher center of gravity and dissimilarity to human movement. We explore curriculum learning with "blinking" — randomly masking sensors and actuators mid-motion — as a method for training robust humanoid policies in simulation. We find that observation blinking fails as a regularizer, as networks compensate through state estimation and revert to the same flawed gait. Actuator blinking, however, physically alters the MDP and forces the development of a more distributed, robust locomotion strategy. These results suggest that effective regularization in physical control requires perturbations that cannot be estimated away.

1 Introduction

Deep RL has enabled amazing advancements in legged robotics, allowing bipedal robots to move across complex environments with high agility. However, deploying these policies in the real world is still heavily constrained with out-of-distribution input and output spaces, since standard deep RL policies rely on perfect and continuous data from sensors and uninterrupted execution from every joint during training. In the real world, robots will almost certainly encounter sensor noise, actuator failures, and environmental obstacles. Although humans can adapt to these changes, such as walking with a limp or moving through the dark, current robot policies experience catastrophic failures when their observation or action spaces are altered. Injecting noise helps, but overcoming these challenges requires a more involved approach, such as implementing hardware dropout.

Although there's prior work on resilience in legged robots, it's mostly focused on stable environments or isolated failure modes. Kim et al. successfully simulated impaired joints with random joint masking, but their approach was limited to quadrupedal robots which have lower center of mass Kim et al. (2024). Furthermore, existing methods treat sensor and actuator dropouts as independent and stochastic noise processes, failing to represent the causal nature of the real world where the environment hazards may actively cause mechanical and sensory failures.

To address these limitations, we introduce the BLIND (Bipedal Locomotion with Intermittent Navigation Data for Environmental Hazards) framework. We aim to train a humanoid to walk while remaining robust to both sensor/actuator dropouts and physical disturbances. We hypothesize that training policies with temporarily missing input and output spaces during training forces resilient, human-like recovery behaviors. Specifically, our work makes three novel contributions to robust humanoid locomotion:

1. **Causal Failure Modeling**: We present a novel simulation environment beyond noise injection, where we couple external physical hazards with internal system failures. In this framework, projectile collisions trigger "blinking" failures, such as proprioceptive sensor, exteroceptive radar, and actuator dropout
2. **Progressive Multi-modal Training**: We design a three-stage curriculum pipeline based around Proximal Policy Optimization (PPO), which safely scales task difficulty by teaching the robot to first walk, then understand its environment, and finally maintain stability while dealing with mechanical and sensory blind spots.
3. **Actuator Masking as a Structural Regularizer**: Through ablations, we present the counter-intuitive empirical finding that intermittent actuator failures act as a regularizer, where forcing local limb limpness alters the Markov Decision Process (MDP) to prevent the neural network from over-relying on specific joints for stability. This reduces fall rates from 24% to 19%, increasing the mean reward.

This ultimately shows a new sim-to-real paradigm where we force policies to survive causal hardware and sensory degradation environments to ensure real-world resilience.

2 Related Work

Sim-to-Real and Standard Domain Randomization

Deploying deep RL policies on legged robots rely on sim-to-real transfer paradigms, and frameworks like Humanoid-Gym demonstrate that high-fidelity physics simulations can produce zero-shot transfer for the bipedal locomotion task Gu et al. (2024). Part of bridging the sim-to-real gap is using domain randomization (DR) by injecting noise into physical parameters like mass, and sensor readings like joint velocity. However, standard DR assumes that the observation and action spaces remain usable even if they're noisy. This doesn't allow policies to prepare for severe and intermittent failures that are possible in the real world, leading to failure when hardware explicitly malfunctions. Our work diverges from standard DR by modeling dropouts as extended changes to the MDP rather than noise.

Fault-Tolerant Control

Recent research has started to explicitly address hardware failures by training policies with impaired joints. For instance, Kim et al. used joint masking to simulate impaired joints and verified that their strategy was successful in allowing a Unitree's Go1 robot to walk in both real-world indoor and outdoor environments with impaired joints. However, their approach was only tested on quadrupedal robots. ((Kim et al., 2024)) Our work adapts this strategy to train the movement of humanoid robots, which presents significant challenges beyond quadrupedal motion.

Additionally, Qiu et al. explore not only joint malfunctions, but also sensor malfunctions Qiu et al. (2025). However, UMC models the dropout as independent and identically distributed stochastic events, whereas in physical reality, failures are causally linked to external environmental stressors such as collisions and impacts.

Causal Hazard Modeling

Existing methods like work by Xu et al. also treat external hazards, such as sudden forces, and internal failures, like blind spots, as separate problems that are independently solved Xu et al. (2025a). Our framework, BLIND, bridges this gap. We argue that developing true resilience requires exposing the policy to a causal chain of damage. Thus, when we simulate physical projectiles that actively hinder the robot, and map these impacts to loss of input data or actuator motion, we force the network to learn unified recovery behaviors. Thus, we tackle the prior shortcoming where the policy experiences and recovers from multimodal, physical trauma.

3 Methods

We pose the bipedal locomotion task with projectile hazards and mechanical failures as a Partially Observable Markov Decision Process (POMDP). Our objective is to train a closed-loop policy π_θ for our 12 degree of freedom humanoid robot (XBot-L) to handle these environmental obstacles that trigger multi-modal dropout. We train our policy using Proximal Policy Optimization (PPO) within the Isaac Gym environment.

3.1 Simulation Dynamics and Low-Level Control

The simulation operates with a physics timestep of $\Delta t_{\text{sim}} = 0.001s$ (1 kHz), and control actions are decimated with a factor of $N_{\text{dec}} = 10$, resulting in a policy frequency of 100 Hz.

The action space consists of target joint position offsets $a_t \in \mathbb{R}^{12}$, which are mapped to physical joint torques via a PD controller:

$$\tau_{t,i} = k_{p,i}(\alpha \cdot a_{t,i} + q_i^{\text{default}} - q_{t,i}) - k_{d,i}\dot{q}_{t,i} \tag{1}$$

with the action scale $\alpha = 0.25$, stiffness $k_{p,i}$, and damping gains $k_{d,i}$ selected to best simulate a real-world humanoid robot. To further facilitate sim-to-real transfer, we employ domain randomization, modeling the executed action $\tilde{a}_t$ as a noisy, temporally smoothed interpolation of the current and previous actions.

3.2 Observation Space and Asymmetric Actor-Critic

We utilize an asymmetric actor-critic architecture to leverage privileged simulation states during training.

Actor observation: The actor receives a 54-dimensional observation o_t at each frame, including information like joint positions and velocities, previous actions, orientation, and our synthetic radar readings. As input, we use an observation horizon of 15 steps, with an input $O_t = [\hat{o}_{t-14}; \ldots; \hat{o}_t] \in \mathbb{R}^{810}$.

Privileged Critic: Our critic calculates the value function $V(\mathbf{c}_t)$ using the privileged observations $\mathbf{c}_t$ with ground-truth data not available to the actor, such as the actual velocity, foot contact booleans, and domain randomization parameters. This helps us accelerate and stabilize training by exploiting the observable nature of the Isaac Gym simulator.

3.3 Synthetic Exteroception and Projectile Dynamics

To test our out-of-distribution robustness, we implement custom procedural projectiles to spawn in our environment. Specifically, we continuously generate 3 kg geometric projectiles at a distance of $d = 2.0$ m away, directed toward the humanoid's CoM and limbs with velocity $v_p = 7.5$ m/s. The policy receives information about the hazards using a custom closest-target radar observation $\mathbf{r}_t \in \mathbb{R}^7$, which isolates the nearest active projectile within a 10 meter radius and represents its relative position and velocity in the robot body frame.

3.4 "Blinking": Causal and Stochastic Masking

A core contribution of BLIND is a *blinking* failure protocol: intermittent, timed dropouts of sensors or actuators that force the policy to develop generalized recovery mechanisms. We define three failure modes:

- **Sensor Dropout (Proprioceptive):** The joint velocity $\dot{q}_{t,j}$ for a randomly selected joint j is zeroed in the actor observation for $D_s = 50$ steps. The motor controller still receives the true value; only the policy is blinded.
- **Actuator Fault (Limpness):** The PD gains of a randomly selected joint are set to zero ($k_{p,j} = k_{d,j} = 0$) for $D_a = 30$ steps, rendering it floppy and unpowered.
- **Exteroceptive Blackout:** The full 7-dimensional radar observation vector is zeroed for $D_e = 50$ steps, blinding the policy to incoming projectiles.

Failures are triggered by two concurrent mechanisms: (1) **stochastic background blinks**, where a new failure in a given mode begins with probability p at each policy step if that mode is currently inactive ($p_s = p_e = 0.01$; $p_a = 0.002$); and (2) **causal, impact-based failures**, where a projectile contact force exceeding $F_{\text{th}} = 10\,\text{N}$ triggers a mode-specific failure: left-leg impacts mask a random joint on indices 0–5, right-leg impacts on 6–11, and torso/head impacts trigger an exteroceptive blackout. If a failure is already active, an impact resets the timer to the full duration D and samples a new random joint.

Failure rate analysis. We model each background failure mode as an alternating renewal process between *Inactive* and *Active* states. Since the waiting time in the Inactive state is geometrically distributed with parameter p, the expected cycle time is $E[T_{\text{cycle}}] = 1/p + D$, and the long-run fraction of time in the failure state is:

$$\text{Fraction Active} = \frac{D}{\frac{1}{p} + D} \tag{2}$$

At the policy frequency of 100 Hz, this yields **33.3%** failure exposure for radar and proprioceptive masking ($50/(100 + 50)$) and **5.66%** for actuator limpness ($30/(500 + 30)$). In a typical 60-second walk, the robot is sensor- or radar-blind for a cumulative $\sim$20 s across $\sim$40 half-second windows, and has a floppy joint for $\sim$3.4 s in $\sim$11 brief 0.3-second bursts — roughly half a gait cycle per occurrence. These regimes are deliberately challenging: the 33% rate prevents reactive last-millisecond dodging, while the 5.66% actuator rate forces rapid weight transfer and dynamic re-balancing on each occurrence.

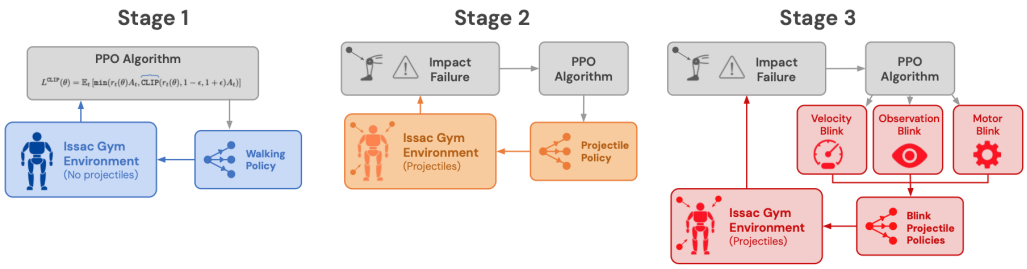


Figure 1: Overview of the BLIND training pipeline, where we use curriculum learning with multi-modal blinking failures to train a humanoid to walk in a projectile-filled environment.

3.5 Curriculum Learning and Reward

Since learning to walk under our environment is highly unstable, we structure the training with a three-stage curriculum:

1. Optimize a baseline walking policy without hazards.
2. Introduce projectiles aiming at the CoM to train evasion and physical recovery.
3. Superimpose the random "blinking" failure modes with the physical hazards.

We use a dense reward function

$$r_t = \Delta t \sum_k w_k \cdot \rho(\mathbf{s}_t, \mathbf{a}_t) \tag{3}$$

which balances positive, Gaussian-shaped terms for velocity tracking, gait phase matching, and base-height maintenance (target of 0.89 m) with quadratic penalty terms for large joint torques and joint accelerations to encourage smooth movement.

We use ELU activations in a MLP with PPO, Generalized Advantage Estimation with discount $\gamma = 0.994$ and decay $\lambda = 0.9$, which we found to yield the highest results for our experiments. We use the clip value $\epsilon = 0.2$.

4 Experimental Setup

We design our experiments to systematically evaluate the role of curriculum stages and the individual/compounded regularizing effects of different blinking modes. Our training and evaluation pipeline is implemented in Isaac Gym Makoviychuk et al. (2021) with the RobotEra XBot-L bipedal humanoid model Gu et al. (2024).

4.1 Curriculum Stages and Resume Hyperparameters

Our training pipeline is structured into three progressive stages to stabilize the policy's gait before exposing it to external hazards and internal failures:

1. **Stage 1: Locomotion Baseline:** The policy is trained from scratch on flat ground without projectiles or blinking failures. This run is trained for 300 iterations (18,000 steps per environment, totaling 73.7M simulation steps) to establish a stable walking gait.
2. **Stage 2: Projectile Resilience:** Resuming from the Stage 1 baseline, we activate the spherical projectile spawning and impact-triggered failures. The model is trained for 100 iterations (6,000 steps per environment, totaling 24.6M simulation steps) to adapt to constant physical perturbations.
3. **Stage 3: Blinking Training:** Resuming from the same Stage 2 checkpoint, the robot is trained under different blinking configurations for 100 iterations (6,000 steps per environment, totaling 24.6M simulation steps) to learn recovery policies under sensory and mechanical malfunctions.

To prevent gait degradation on curriculum transfer, we apply three hyperparameter overrides: a fixed learning rate ($\eta = 5 \times 10^{-5}$), an exploration noise reset ($\sigma = 0.3$), and discarding the Adam optimizer state. Full details are in Appendix A.

4.2 Stage 3 Experimental Branches

In Stage 3, we design five parallel branches starting from the same Stage 2 checkpoint (iteration 100) to evaluate the individual and joint effects of sensor, actuator, and exteroceptive blinking:

- **Branch A (Control):** Projectiles remain active, but all blinking failure modes are disabled. This evaluates the performance of a policy that undergoes direct continuation of Stage 2.
- **Branch B (Sensor Blink):** Enables stochastic background proprioceptive joint velocity masking ($p_s = 0.01$, $D_s = 50$). Actuator limpness and radar blackouts are disabled.
- **Branch C (Actuator Blink):** Enables stochastic background joint actuator limpness ($p_a = 0.002$, $D_a = 30$). Sensor masking and radar blackouts are disabled.
- **Branch D (Radar Blink):** Enables stochastic background exteroceptive radar masking ($p_e = 0.01$, $D_e = 50$). Actuator limpness and proprioceptive sensor masking are disabled.
- **Branch E (Combined Blinking):** All three stochastic background blinking failure modes are simultaneously enabled (proprioceptive sensor masking, actuator limpness, and radar blackout).

All five branches are trained for exactly 100 iterations under the same fixed learning rate ($\eta = 5 \times 10^{-5}$) and batch size to ensure a mathematically fair comparison.

4.3 Evaluation Protocol

To benchmark policy robustness, we run offline evaluation sweeps using a custom evaluation module. For each evaluated checkpoint, we collect data over $N_{\text{eval}} = 100$ independent trials. The policy runs in inference mode ($\mathbf{a}_t \sim \pi_\theta(\mathbf{s}_t)$ with no exploration noise) for a maximum of 1,000 steps (10.0 s) per episode. We report the following three metrics across all Stage 3 branches:

- **Mean Total Reward:** The cumulative sum of rewards $\sum_t r_t$ accumulated over a full episode, reflecting both locomotion quality and projectile resilience.
- **Mean Survival Time:** The average wall-clock duration (in seconds) the robot remains upright before falling, where termination is triggered when base height drops below 0.40 m or base tilt exceeds roll/pitch thresholds.
- **Episode Fall Rate:** The percentage of the 100 evaluation trials that ended in a fall before the 1,000-step (10.0 s) timeout.

5 Results

5.1 Quantitative Evaluation

To rigorously assess the efficacy of the BLIND framework, we evaluate our learned policies across multiple multi-modal failure conditions. Our primary objective is to quantify how the coupled masking affects the biped's ability to navigate hazards.

We evaluate each trained policy using the same seed across 100 different episodes. The evaluation script records per-episode statistics including Mean Total Reward, Mean Survival Time, and Episode Fall Rate. The primary robustness metric is mean episode length, which measures how long the robot can continue walking before failure. Since the simulation control timestep is 0.01 seconds, episode length can also be converted into survival time in seconds.

5.2 Primary Results

We measure policy performance using the mean episodic return, mean survival time (s), and episode fall rate to ensure both efficient and stable locomotion in our task.

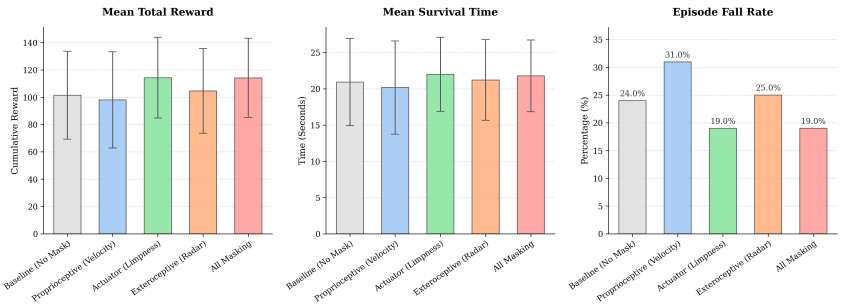
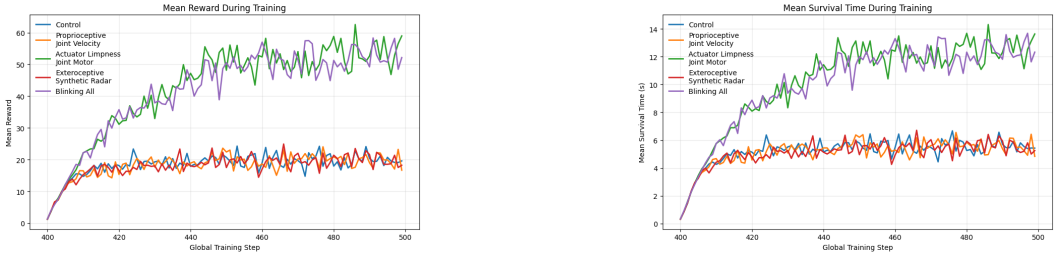


Figure 2: Results of our five different dropout methods evaluated against the mean reward, fall rate, and episode length

The most significant empirical finding of our study is the counter-intuitive performance of actuator masking. Although intuition suggests that randomly disabling joint motors to force limpness would destabilize our humanoid system, our experiments demonstrate that actuator limpness acts as a powerful regularizer. Comparing the baseline control policy (projectiles, no masking) to the actuator-masked policy, the actuator dropouts reduces the fall rate from 24% to 19%. Furthermore, physically changing the MDP forces a more robust gait, increasing the mean reward from 101.45 to 114.32. These are shown in Figure 2.



(a) Mean Reward During Training

(b) Mean Survival Time During Training

Figure 3: Mean Metrics During Training

During training, the actuator limpness masking condition exhibited a noticeably delayed plateau in mean reward compared to the other masking paradigms. This suggests that the actuator masking task encouraged a more adaptive locomotion strategy. In particular, the humanoid appears to have learned to compensate for joint motor impairment by developing more dynamic whole-body motion, rather than relying on rigid or conservative postures that may temporarily preserve balance but ultimately lead to termination of the episode.

Conversely, masking the sensory observation spaces without altering the physical dynamics results in a degraded survival time of 20.18 seconds (from 20.932 s) and high fall rate of 31% due to its reliance on velocity feedback for stability without being able to compensate for losing its internal observations.

5.3 Ablation Study

To disentangle individual and compounded effects of our failure modalities, we also run an exhaustive ablation over 5 notable combinations of our masking combinations: Baseline, Sensor Dropout, Actuator Limpness, Radar Blackout, and Fully Coupled, across two distinct hyperparameter sweeps:

1. Low-intensity sweep ($0.5 \times$ Default): $p_s = 0.0025, p_a = 0.0005, p_r = 0.0025$
2. High-intensity sweep ($2 \times$ Default): $p_s = 0.01, p_a = 0.002, p_r = 0.01$

The sensitivity analysis shows that for proprioceptive masking, using a lower dropout rate ($0.5\times$) results in a higher but manageable fall rate of 27.5%, but doubling the dropout rate causes a catas-

Table 1: Ablation of Coupled Masking Across Varying Dropout Intensities

Configuration	Fall Rate (%)		Mean Return	
	0.5× Default	**2.0× Default**	**0.5× Default**	**2.0× Default**
Baseline	24.0	24.0	101.45	101.45
Velocity-Only	27.5	38.2	94.20	72.15
Actuator-Only	21.0	20.5	108.60	112.40
Radar-Only	24.8	28.5	99.10	92.30
Coupled All	22.1	23.4	106.80	102.50

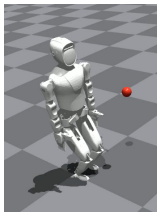
trophic collapse of 38.2% fall rate and only 72.15 mean return, suggesting the state estimation isn't able to bridge the larger temporal gaps in joint velocity feedback.

Furthermore, the actuator-only configuration confirms that physical masking acts as regularizer; we see that the high intensity sweep yields marginal improvement in the fall rate (drops the fall rate by 0.5%), while boosting the mean return. This means that the regularization results in a more resilient baseline gait, but this is only to a certain extent before seeing diminishing returns.

5.4 Qualitative Analysis

Qualitative analysis of the Isaac Gym renderings reveals notable behavioral differences in the actuator-blinked policy from the baseline. Rather than adopting full strides, the humanoid learns a shuffling gait that distributes weight more evenly and reduces reliance on any single joint during movement, which is a direct behavioral consequence of being forced to move with randomly disabled actuators during training. Specifically, the robot moves with essentially two steps per leg before switching; an emergent behavior which reduces the vulnerable motion of shifting weight between the legs. Additionally, the policy develops an impact recovery behavior: upon projectile contact, the humanoid briefly interrupts forward movement to stabilize before resuming. This two-phase response mirrors how humans naturally recover from unexpected physical perturbations.

(a) Robot policies trained without environmental projectiles walk with larger yet less stable strides by standing straighter



(b) Introducing projectiles with actuator and sensor dropout results in the robot bending down to not immediately fall upon impact

Figure 4: Comparison of different walking strategies throughout the different stages

Comparing the policies trained before and after introducing the projectiles, we see that as the policy continues to adapt to the environmental hazards during stage 2 of the curriculum, it learns behaviors to overcome the new challenge of the projectiles. Although the humanoid originally maximized its vertical distance by standing with unbent leg joints, enabling it to cover more horizontal distance with each step forward, this configuration was quite fragile and was easily knocked over by all projectiles. After 20 iterations of stage 2 training, the new humanoid's lower CoM reduced the pushback from the projectiles, minimizing the penalties from moving backward, while also reducing the likelihood of collapsing due to joint failure.

6 Discussion

The results of the BLIND framework show a novel finding in the training of humanoid policies: although sensory and exteroceptive masking fails to improve performance, intermittent actuator masking acts as a strong regularizer.

Actuator Dropout as a Physical Regularizer

Standard DR and noise injection often fail to prevent policies from converging into local optima, such as a rigid, straight-legged walk that we observed in our baseline. The stage 3 training curves show that although the control, proprioceptive, and radar policies plateau at a mean survival time of 5 to 6 seconds, the actuator limpness and combined blinking policies break this plateau to converge at nearly 14 seconds. We hypothesize actuator masking acts as a "physical" dropout mechanism to escape the rigid walk developed during stage 1 without the projectiles. Thus, the actuator-dropout policies are able to fundamentally recover and succeed in the projectile environment, whereas the worse-performing policies are still relying on a fragile walking pattern unsuited for the projectiles.

Information vs Physical Bottleneck

Our experiments showed a major distinction between information robustness from the sensors, and mechanical robustness from the actuators. When proprioceptive (velocity) or exteroceptive (radar) data is masked, the MDP's physical dynamics remain unchanged. The actor network, which has finite representational capacity, likely tries to estimate the 50-step temporal gap, but the high fall rate and degraded mean return suggest the network tends to hallucinate the missing variables to try and maintain its baseline walk.

On the other hand, the actuator limpness acts as a physical bottleneck, where the network can't "estimate" a solution to a dead motor. Therefore, it compensates by relying other degrees of freedom and distributing its control, where the load of balancing and locomotion is shared across the entire body. This forces the policy to continuously redistribute its effort, resulting in more resilient behaviors.

Synergistic Effects in Combined Blinking

We see that although the just masking the sensor data is detrimental, its negative effects are largely mitigated when coupled with actuator masking in the "Combined Blinking" configuration. This produces the same high survival times of the Actuator-only policy during training (above 13 seconds) and matches its lowest fall rate during evaluation. This suggests a hierarchical dependence in training policies, where the mechanical adaptation is a foundation where sensory adaptation is built upon. Only after the actuator-blinking training forces the humanoid into a stable, low CoM gait does the policy become more tolerant to the gaps from sensor masking and survive the sensory blackouts.

Ultimately, these findings challenge the existing sim-to-real transfer paradigm of closing the sensory reality gap using noise and latency. Our results show that preparing for mechanical failures results in a more robust baseline than preparing only for noisy data, especially since real-world robots are highly susceptible to thermal throttling and unexpected contact forces. Thus, embedding hardware dropout into the training curriculum through the BLIND framework demonstrates that forcing robots to survive mechanical failures is a highly effective method for hazard-resilient locomotion.

7 Conclusion

We investigated curriculum learning with blinking as a method for improving humanoid robustness in simulation, progressively introducing projectile hazards and sensory/actuator masking across three training stages. Actuator blinking proved to be an effective regularizer, preventing over-reliance on specific joints and reducing catastrophic fall rates from 24% to 19% in hazardous environments, while observation blinking failed to meaningfully alter learned behavior. Together these results suggest that effective regularization in physical control requires perturbations at the level of the MDP rather than the observation space.

Future Work: Several directions remain open. First, our blinking protocol masks a single joint or modality at a time; extending to simultaneous multi-joint or multi-modal dropout could expose the policy to more realistic compound failure scenarios. Second, our causal failure model ties projectile impacts to fixed-duration dropouts; a learned or adaptive failure model that conditions dropout duration and severity on impact force could produce more physically grounded training signal. Third, all experiments are conducted in simulation—sim-to-real transfer of the actuator-blinked policy to physical hardware (e.g., XBot-L) remains an important validation. Finally, our synthetic radar could be replaced with a real depth or RGB camera, enabling blinking of actual perceptual inputs and closing the gap between our simulated exteroceptive blackouts and real-world sensor failures.

8 Team Contributions

- **Iris Xu:** Initial pipeline experimentation with MuJoCo and Unitree G1, implementing and running actuator limpness and synthetic radar blinking on XBot-L model, writing and running evaluation script
- **Jamin Xie:** Initial pipeline experimentation with end-to-end MuJoCo, implementing projectiles and joint/camera failures, implementing and debugging training and experiment pipelines, running 3-stage curriculum training
- **Eric Liang:** Set up Google Colab with Humanoid-Gym environment, implemented projectiles, run 3-stage curriculum training and ablation experiments, rendered rollouts

Changes from Proposal We initially planned to experiment using a Unitree G1 humanoid model and MuJoCo, but through various experimentation by group members we found this pipeline was difficult to work with using Modal and that we could not train our intended experiments. We switched to using IsaacGym with RobotEra's XBot-L humanoid model instead.

We had also originally planned to mask input camera frames, but found that the RobotEra repository did not use camera feed as an input to the PPO policy. Thus, we had to experiment with blinking for synthetic radar masking instead.

Our team contributions were adjusted between the proposal and final submission based on compute availability and figuring out which implementation frameworks to use.

References

Lukas Biewald. 2020. Experiment Tracking with Weights and Biases. `https://www.wandb.com/`.

DeepMind. 2022. MuJoCo Menagerie: A collection of high-quality simulation models for MuJoCo. `https://github.com/google-deepmind/mujoco_menagerie`.

Xinyang Gu, Yen-Jen Wang, Xiang Zhu, Chengming Shi, Yanjiang Guo, Yichen Liu, and Jianyu Chen. 2024. Humanoid-Gym: Reinforcement Learning for Humanoid Robot with Zero-Shot Sim2Real Transfer. *arXiv preprint arXiv:2404.05695* (2024).

Mincheol Kim, Ukcheol Shin, and Jung-Yup Kim. 2024. Learning Quadrupedal Locomotion with Impaired Joints Using Random Joint Masking.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. 2021. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155 [cs.CL]

ProfessorNova. [n. d.]. PPO Humanoid. `https://github.com/ProfessorNova/PPO-Humanoid`.

Yu Qiu, Xin Lin, Jingbo Wang, Xiangtai Li, Lu Qi, and Ming-Hsuan Yang. 2025. UMC: A Unified Approach for Resilient Control of Legged Robots Across Masked Malfunction Training. Available at: https://arxiv.org/html/2502.03035v1.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. 2021. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research* 22, 268 (2021), 1–8.

Unitree Robotics. [n. d.]. Unitree RL Gym. `https://github.com/unitreerobotics/unitree_rl_gym`.

Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 5026–5033.

Mark Towers, J. K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. 2023. Gymnasium. `https://zenodo.org/record/8127025`.

Xiaomeng Xu, Dominik Bauer, and Shuran Song. 2025a. RoboPanoptes: The All-Seeing Robot with Whole-body Dexterity. In *Proceedings of Robotics: Science and Systems*. Los Angeles, CA, USA. doi:`10.15607/RSS.2025.XXI.042`

Xiaomeng Xu, Dominik Bauer, and Shuran Song. 2025b. RoboPanoptes: The All-Seeing Robot with Whole-body Dexterity. Available at: https://robopanoptes.github.io/.

A Implementation Details

Curriculum Transfer Hyperparameters. Naively resuming a converged PPO checkpoint into a new environment stage causes two failure modes. First, the adaptive KL-divergence learning rate schedule detects a near-zero initial KL (because the policy parameters are already structured) and repeatedly multiplies the learning rate by 1.5, causing it to explode from 10^{-5} to $> 10^{-3}$ within 20 iterations and destroying the pre-trained gait. We override this by fixing $\eta = 5 \times 10^{-5}$ for all Stage 2 and Stage 3 resumes. Second, by the end of Stage 1 the exploration noise standard deviation σ has decayed from 1.0 to ≤ 0.2, starving Stage 2/3 of the exploration needed to learn new behaviors (dodging, weight-shifting). We reset $\sigma = 0.3$ on each curriculum transfer. Finally, we discard the Adam optimizer's accumulated first- and second-moment vectors (`load_optimizer = False`) when entering a new stage, since the observation/action distributions shift with the introduction of projectiles or blinking; for intra-stage continuations we retain the optimizer state (`load_optimizer = True`).

Computing Infrastructure. All models are trained and evaluated in Google Colab instances using NVIDIA A100 GPUs. The vectorized environment runs with $4,096$ parallel simulated robots, mapping environment updates directly to GPU memory using Isaac Gym's CUDA tensor API. Training metrics are tracked and logged using Weights & Biases Biewald (2020).